



Titre: Simulation d'écoulements turbulents sur profils d'ailes au moyen de
Title: la méthode de Boltzmann sur réseau

Auteur: Nicolas Pellerin
Author:

Date: 2017

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Pellerin, N. (2017). Simulation d'écoulements turbulents sur profils d'ailes au
Citation: moyen de la méthode de Boltzmann sur réseau [Ph.D. thesis, École Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/2515/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/2515/>
PolyPublie URL:

**Directeurs de
recherche:** Marcelo Reggio
Advisors:

Programme: Génie mécanique
Program:

UNIVERSITÉ DE MONTRÉAL

SIMULATION D'ÉCOULEMENTS TURBULENTS SUR PROFILS D'AILES AU MOYEN
DE LA MÉTHODE DE BOLTZMANN SUR RÉSEAU

NICOLAS PELLERIN

DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIAE DOCTOR
(GÉNIE MÉCANIQUE)

AVRIL 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

SIMULATION D'ÉCOULEMENTS TURBULENTS SUR PROFILS D'AILES AU MOYEN
DE LA MÉTHODE DE BOLTZMANN SUR RÉSEAU

présentée par : PELLERIN Nicolas

en vue de l'obtention du diplôme de : Philosophiae Doctor

a été dûment acceptée par le jury d'examen constitué de :

M. LAURENDEAU Éric, Ph. D., président

M. REGGIO Marcelo, Ph. D., membre et directeur de recherche

M. TRÉPANIÉ Jean-Yves, Ph. D., membre

M. MALASPINAS Orestis, Ph. D., membre externe

DÉDICACE

À Noah,

*parce que tu mets de la joie dans ma vie
et que ta présence a permis de reléguer
le travail et les études au second
rang de mes priorités.*

REMERCIEMENTS

Je tiens à remercier mon directeur de recherche, le professeur Marcelo Reggio, pour avoir pris la chance de m'accepter comme étudiant malgré le risque que ça impliquait, étant donné ma situation particulière où j'avais comme objectif de combiner travail et études. Je vous remercie aussi de toujours avoir été présent pour m'aider et répondre à mes questions, et d'avoir accepté que nous collaborions à distance. Vous m'avez aussi toujours supporté, même dans les moments d'épuisement, ce que j'ai grandement apprécié.

Je remercie aussi Sébastien Leclaire, mon ancien collègue de Polytechnique et maintenant chercheur, pour tous les conseils qu'il m'a prodigués. À plusieurs reprises tu as fait en sorte que j'évalue mes travaux selon la perspective d'un chercheur plutôt que de celle d'un ingénieur, ce qui était un réflexe naturel pour moi. J'ai aussi beaucoup apprécié nos échanges d'idées et avoir pu contribuer à certaines de tes publications.

Je désire aussi remercier Mylène Leroux, qui à maintes reprises a presque agi comme co-directrice de recherche. Tes conseils et encouragements m'ont permis de prendre du recul et de garder une perspective globale sur mes travaux. Tu as toujours su me rappeler que le doctorat n'était pas une finalité en soi et qu'il était important de prendre des temps d'arrêt afin de se ressourcer. Je t'en remercie.

RÉSUMÉ

La méthode de Boltzmann sur réseau n'a cessé de gagner en popularité au cours des vingt-cinq dernières années en tant qu'alternative aux méthodes existantes de dynamique numérique des fluides. De nombreux travaux de recherches sont publiés à chaque année et assurent un développement continu de la méthode. L'éventail de types d'écoulements de fluides qu'elle permet de simuler s'est aussi étendu et n'est pas restreint aux écoulements laminaires à une seule phase. C'est dans cette optique que les travaux de cette thèse s'inscrivent, car la méthode est appliquée au cas spécifique des écoulements turbulents sur profils d'ailes. Les travaux présentés contribuent donc à étendre et valider la méthode pour une application d'intérêt en ingénierie.

L'objectif principal de cette thèse est de développer un programme de simulation d'écoulements turbulents au moyen de la méthode de Boltzmann sur réseau, offrant la particularité d'utiliser un réseau pouvant assurer un raffinement local. Afin de remplir cet objectif, cette recherche est divisée en trois parties. La première consiste à développer et valider une nouvelle méthode de conditions aux frontières courbes plus versatile que celles disponibles dans la littérature. Dans la deuxième partie, un programme de simulations d'écoulements turbulents est développé afin de combler certaines lacunes identifiées dans la littérature. Ce programme combine une configuration de réseau de type multi-domaines, l'opérateur de collision en cascade et la résolution de l'équation différentielle du modèle de turbulence Spalart-Allmaras. Cette approche combinée n'a pas été proposée antérieurement dans la littérature. Le nouveau modèle de conditions aux frontières est aussi inclus dans le programme. La troisième partie de la recherche consiste à développer un programme basé sur un réseau de type non-structuré, présentant l'avantage d'offrir la possibilité d'utiliser un raffinement directionnel. Le programme inclut aussi le modèle de turbulence Spalart-Allmaras, et fournit ainsi une approche globale n'ayant encore une fois pas de précédent dans la littérature touchant la méthode de Boltzmann sur réseau.

Trois articles scientifiques, fournis en annexe, ont été rédigés à partir des travaux exécutés pour chaque partie de la recherche. Les résultats qui y sont présentés indiquent que la méthode de conditions aux frontières est versatile et applicable particulièrement aux écoulements à hauts nombres de Reynolds. Ces résultats montrent aussi que les programmes de simulation d'écoulements turbulents produisent des caractéristiques aérodynamiques concordant avec des valeurs de référence et que le raffinement de réseau en constitue un facteur essentiel.

ABSTRACT

Over the last twenty-five years, the lattice Boltzmann method's popularity as an alternative to existing computational fluid dynamics methods hasn't stop growing. Abundant research is published every year and ensure the continued development of the method. The spectrum of fluid flows that it allows to simulate has also increased and is not restricted to single phase laminar flows. In fact, the research work of this thesis consists of applying the method to turbulent flows around airfoils. Therefore, the works presented herein contribute to the broadening and validation of the method for a specific and important engineering application.

The main objective of this thesis is to develop a computer program for simulating turbulent airfoil flows that is based on the lattice Boltzmann method and incorporates the possibility of using local lattice refinement. In order to achieve this objective, the research is divided in three parts. The first one consists of the development and validation of a new curved boundary condition method that is more versatile than those available in the literature. In the second part, a computer program for simulating turbulent flows is developed with the aim of filling gaps identified in the literature. This program combines a multi-domain lattice configuration, the cascaded collision operator, and a solving methodology for the differential equation from the Spalart-Allmaras turbulence model. This combined approach has not been proposed before in the literature. The new boundary condition method is also included in the program. The third part of the research consists of the development of a computer program based on an unstructured lattice, which provides the advantage of using directional refinement. The program also includes the Spalart-Allmaras turbulence model. Overall it provides an approach that has not been proposed either in the lattice Boltzmann literature.

Three scientific articles were written based on the work executed for each part of the research. They are available in the Appendix. The results presented in these articles show that the boundary condition method is versatile and particularly applicable to high Reynolds number flows, that the computer programs for simulating turbulent airfoil flows produce aerodynamic characteristics that are in agreement with reference data, and that lattice refinement constitutes a key element to solution accuracy.

TABLE DES MATIÈRES

DÉDICACE.....	III
REMERCIEMENTS	IV
RÉSUMÉ.....	V
ABSTRACT	VI
TABLE DES MATIÈRES	VII
LISTE DES TABLEAUX.....	X
LISTE DES FIGURES	XI
LISTE DES SIGLES ET ABRÉVIATIONS	XII
LISTE DES ANNEXES	XIII
CHAPITRE 1 INTRODUCTION.....	1
1.1 Cadre théorique	2
1.1.1 Théorie cinétique des gaz.....	2
1.1.2 Automates cellulaires de gaz sur réseau.....	6
1.1.3 Automates cellulaires et équation de Boltzmann	14
1.1.4 Méthode de Boltzmann sur réseau	20
1.1.5 Évolution de la méthode de Boltzmann sur réseau	27
1.2 Problématique.....	34
1.2.1 Question de recherche	34
1.2.2 Lacunes actuelles.....	35
1.2.3 Hypothèse et buts	36
1.2.4 Objectifs	36
1.2.5 Nouveautés	38
CHAPITRE 2 REVUE CRITIQUE DE LA LITTÉRATURE	40

2.1	Écoulements turbulents sur ailes	40
2.2	Conditions aux frontières sur surface solide courbe	46
2.2.1	Approche générale par réflexion	47
2.2.2	Approche volumétrique.....	50
2.2.3	Approche immergée	51
2.2.4	Approche versatile.....	52
CHAPITRE 3 SYNTHÈSE DE L'ENSEMBLE DU TRAVAIL		54
3.1	Nouvelle méthode de conditions aux frontières	54
3.1.1	Description de la méthode.....	55
3.1.2	Validation	58
3.1.3	Conclusions	62
3.2	Écoulements turbulents sur profils d'ailes avec réseaux multi-domaines.....	63
3.2.1	Développement du programme	64
3.2.2	Validation	69
3.2.3	Conclusions	73
3.3	Écoulements turbulents sur profils d'ailes avec réseaux non-structurés	75
3.3.1	Développement du programme	75
3.3.2	Validation	80
3.3.3	Conclusions	87
3.3.4	Considération additionnelle.....	88
CHAPITRE 4 DISCUSSION GÉNÉRALE		90
4.1	Modèle de conditions aux frontières versatile.....	90
4.2	Opérateur de collision stable et réseau multi-domaines.....	91
4.3	Modèle de turbulence Spalart-Allmaras et réseau multi-domaines	92

4.4	Écoulements turbulents, LBM et réseaux non-structurés.....	93
CHAPITRE 5 CONCLUSION ET RECOMMANDATIONS		95
5.1	Recommandations	97
RÉFÉRENCES.....		99
ANNEXES		110

LISTE DES TABLEAUX

Tableau 1.1 : Types de collisions du modèle FHP.....	12
Tableau 1.2 : Valeur des poids pour les distributions à l'équilibre.	19
Tableau 3.1 : Coefficients aérodynamiques obtenus au moyen du programme multi-domaines. .	70
Tableau 3.2 : Coefficients aérodynamiques obtenus au moyen du programme pour réseau non-structuré.....	82
Tableau 3.3 : Coefficients aérodynamiques obtenus pour l'angle d'attaque de 12° au moyen du réseau raffiné à $1/32768^\circ$ de corde et des deux modèles de turbulence.	83

LISTE DES FIGURES

Figure 1.1 : Exemple d'évolution d'automate cellulaire selon le Jeu de la vie de Conway. L'ordre chronologique des images est de gauche à droite et de haut en bas, représentant 16 pas de temps successifs.	7
Figure 1.2 : Schéma du réseau pour le modèle HPP.	9
Figure 1.3 : Schéma du réseau pour le modèle FHP.	11
Figure 1.4 : Schéma du réseau d2q9.	21
Figure 3.1 : Schémas pour l'application des conditions aux frontières selon a) Guo et al. [28] et b) la nouvelle méthode proposée. Les cercles pleins représentent des nœuds fluides et les cercles vides représentent des nœuds solides.	56
Figure 3.2 : Représentation du réseau non-structuré et des points d'interpolations.	78
Figure 3.3 : Illustration du réseau près du bord d'attaque pour le raffinement de $1/32768^\circ$ de corde et l'angle d'attaque de 12°	84
Figure 3.4 : Profils de coefficients de pression obtenus pour l'angle d'attaque de 12° au moyen du réseau raffiné à $1/32768^\circ$ de corde et des deux modèles de turbulence.	85
Figure 3.5 : Profils de vitesse près du bord d'attaque obtenus pour l'angle d'attaque de 12° au moyen du réseau raffiné à $1/32768^\circ$ de corde et des deux modèles de turbulence.	86
Figure B.1 : Avis d'acception par les éditeurs de l'article «Equilibrium distributions for straight, curved, and immersed boundary conditions in the lattice Boltzmann method».	122
Figure C.1 : Avis d'acception par les éditeurs de l'article «An implementation of the Spalart-Allmaras turbulence model in a multi-domain lattice Boltzmann method for solving turbulent airfoil flows».	144
Figure D.1 : Avis d'acception par les éditeurs de l'article «Solving incompressible fluid flows on unstructured meshes with the lattice Boltzmann method».	172

LISTE DES SIGLES ET ABRÉVIATIONS

BGK	Bhatnagar, Gross et Krook
CFD	Dynamique numérique des fluides (Computational Fluid Dynamics)
CFL	Condition Courant–Friedrichs–Lewy
DNS	Simulation numérique directe (Direct Numerical Simulation)
ELBM	Méthode de Boltzmann sur réseau entropique (Entropic Lattice Boltzmann Method)
FVLBM	Méthode de Boltzmann par volume finis (Finite Volume Lattice Boltzman Method)
GPU	Processeur graphique (Graphics Processing Unit)
LBM	Méthode de Boltzmann sur réseau (Lattice Boltzmann Method)
LES	Simulation des grandes échelles (Large Eddy Simulation)
LGCA	Automates cellulaires de gaz sur réseau (Lattice Gas Cellular Automata)
RANS	Équations moyennées de Navier-Stokes (Reynolds-Averaged Navier-Stokes equations)
RLBM	Méthode de Boltzmann sur réseau régularisée (Regularized Lattice Boltzmann Method)
S-A	Spalart-Allmaras
Re	Nombre de Reynolds

LISTE DES ANNEXES

Annexe A – La méthode de Boltzmann sur réseau et les équations de Navier-Stokes.....	110
Annexe B – ARTICLE 1 : EQUILIBRIUM DISTRIBUTIONS FOR STRAIGHT, CURVED, AND IMMERSED BOUNDARY CONDITIONS IN THE LATTICE BOLTZMANN METHOD	122
Annexe C – ARTICLE 2 : AN IMPLEMENTATION OF THE SPALART-ALLMARAS TURBULENCE MODEL IN A MULTI-DOMAIN LATTICE BOLTZMANN METHOD FOR SOLVING TURBULENT AIRFOIL FLOWS	144
Annexe D – ARTICLE 3 : SOLVING INCOMPRESSIBLE FLUID FLOWS ON UNSTRUCTURED MESHES WITH THE LATTICE BOLTZMANN FLUX SOLVER	172

CHAPITRE 1 INTRODUCTION

Ce projet de thèse doctorale fut à l'origine inspiré par l'idée d'étendre les champs d'applications d'une méthode de simulation numérique dont la popularité est en croissance, soit la méthode de Boltzmann sur réseau ou LBM, au problème de simulation de formation de glace sur des objets tels les câbles électriques ou les ailes d'avions. La piste de recherche initiale consistait à déterminer la faisabilité d'utiliser la LBM à la fois pour simuler un écoulement aérodynamique et pour modéliser le transport de gouttelettes d'eau dans l'écoulement d'air. De plus, le maillage structuré, ou plutôt le réseau de la LBM apparaissait avantageux pour le traitement des changements de géométrie dus à l'accumulation de glace, parce qu'on pouvait anticiper ne pas devoir procéder à de coûteuses refontes de maillages entre chaque couche de glace accumulée. Or, peu de temps après le début du processus de recherche, il s'est avéré clair que la simulation d'écoulements aérodynamiques représentait une avenue de recherche qui à elle seule fournissait suffisamment de défi pour le présent travail, puisque certaines considérations nécessitaient encore un effort de développement et de validation. Plus précisément, la simulation d'écoulements turbulents autour de profils d'ailes d'avions au moyen de la LBM avait été traitée dans la littérature scientifique mais certaines lacunes étaient présentes, particulièrement pour le type de maillage utilisé et la qualité de certains résultats. De plus, l'implémentation de modèles de turbulence n'était pas toujours décrite avec un niveau de détail suffisant.

C'est dans ce contexte que la nature du présent travail s'est précisée et que le sujet a été fixé sur l'utilisation de la LBM pour la simulation d'écoulements turbulents autour de profils d'ailes. Puisque la LBM constitue, tel que mentionné plus haut, une méthode dont l'utilisation est croissante, et ce dans divers domaines d'applications, cette thèse doctorale s'inscrit donc de façon plus générale dans l'ensemble des travaux qui la font avancer et lui permettent de représenter une approche alternative viable aux méthodes de simulation plus établies. Il est donc approprié de débiter le présent travail par une description de la LBM, en expliquant son origine et son développement, ce qui fournira le cadre théorique de la recherche. Cependant, puisque l'objectif poursuivi consiste à appliquer la LBM à un problème pratique d'ingénierie plutôt que d'étudier ses fondements théoriques, de longues dérivations mathématiques ne seront pas reproduites. Les références pertinentes seront fournies au lecteur qui désire plus de détails.

1.1 Cadre théorique

La LBM trouve son origine dans la théorie cinétique des gaz et les automates cellulaires de gaz sur réseau. Elle consiste à modéliser le milieu continu d'un fluide au moyen de l'évolution de distributions de densité de particules sur un réseau structuré. Ces distributions, ou conceptuellement les particules qu'elles représentent, se déplacent et entrent en collision selon des paramètres spécifiques permettant de reconstituer adéquatement les propriétés macroscopiques du fluide. C'est une approche probabiliste, car les distributions de densité de particules ne représentent pas chaque molécule d'un fluide de façon individuelle, mais plutôt un ensemble de molécules qu'il est probable de trouver à un instant et un lieu donnés, se dirigeant dans une direction et à une vitesse communes selon la configuration du réseau. À priori cette description sommaire peut sembler très abstraite mais la LBM constitue en réalité une méthode plutôt simple. Une description des origines et du développement de cette méthode permet d'en saisir les principes de façon plus concrète. La description de la méthode commencera par un rappel de la théorie cinétique des gaz, puis l'introduction de cette théorie dans les automates cellulaires de gaz sur réseau sera discutée, car c'est ce qui a initié l'évolution de la LBM vers les modèles que nous utilisons aujourd'hui. Ensuite, les équations de la LBM sur lesquelles est basé le présent travail seront formalisées.

1.1.1 Théorie cinétique des gaz

Lorsqu'on veut étudier le comportement de la matière, on peut observer cette dernière de plusieurs façons, dépendamment du point de vue. À priori, les solides, les liquides ou les gaz nous apparaissent comme des milieux continus dans lesquels la matière occupe tout l'espace. C'est la perspective macroscopique. Au contraire, si on arrivait à observer individuellement les particules qui composent la matière, que soit des atomes ou des molécules, on aurait une perspective microscopique. On constaterait rapidement aussi qu'il serait impossible d'étudier le mouvement et les interactions de toutes les particules afin d'en déduire le comportement global de la matière. Or, on peut aussi regarder cette dernière d'une façon conceptuelle, à une échelle se trouvant entre les perspectives microscopique et macroscopique, en prenant une approche statistique. C'est l'échelle mésoscopique, pour laquelle on s'intéresse au comportement probable de l'ensemble des particules plutôt qu'aux mouvements des particules individuelles. La théorie cinétique des gaz s'inscrit dans cette troisième perspective.

Cette théorie vise à relier les propriétés macroscopiques d'un gaz aux caractéristiques des mouvements des particules qui le composent. Un volume donné de gaz contient un très grand nombre de particules qui se déplacent très rapidement et dont le mouvement est aléatoire en raison des collisions qui se produisent entre les particules et contre les parois du volume, et du fait que leur état initial est indéterminé [1]. La pression du gaz est mesurée au moyen de l'échange de quantité de mouvement lors des collisions entre les particules et les parois, tandis que la température est reliée à l'énergie cinétique des particules. Or, il n'est pas nécessaire de connaître la vitesse individuelle de chacune des particules dans un volume donné pour déterminer les propriétés macroscopiques, car en raison de leur mouvement aléatoire il ne suffit que de connaître la vitesse moyenne [1]. La théorie cinétique des gaz est fondée sur les hypothèses suivantes [2,3]:

- le volume occupé par toutes les particules est négligeable en comparaison du volume occupé par le gaz, ce qui revient aussi à dire que la distance entre les particules est grande en comparaison de la dimension de ces dernières,
- toutes les particules ont la même masse et sont représentées par des sphères,
- le nombre de particules est suffisamment élevé pour qu'on puisse les considérer de façon statistique, c'est-à-dire qu'on puisse utiliser leurs propriétés moyennes pour établir les propriétés macroscopiques du gaz,
- les particules se déplacent constamment et rapidement, de façon aléatoire,
- les collisions entre les particules et contre les parois sont parfaitement élastiques,
- à part les collisions, toutes autres interactions entre les particules sont négligeables,
- la durée des collisions est très courte en comparaison du temps passé par les particules à se déplacer entre les collisions,
- le mouvement des particules obéit aux lois de Newton.

Selon ces hypothèses et en particulier l'absence d'interactions entre les particules autres que les collisions, la théorie s'applique aux gaz parfaits [2,4]. Pour de tels gaz se trouvant dans un état d'équilibre thermodynamique, la vitesse probable des particules est fournie par la distribution de Maxwell-Boltzmann [5]:

$$f(v) = \left(\frac{m}{2\pi k_B T} \right)^{\frac{3}{2}} 4\pi v^2 e^{\left(\frac{-mv^2}{2k_B T} \right)} \quad (1.1)$$

où v est la vitesse (scalaire) d'une particule, m représente sa masse, k_B est la constante de Boltzmann et T est la température du gaz. Cette loi représente la distribution de probabilité d'une particule de se déplacer à une vitesse v . Or, si le nombre total de particules contenues dans un volume est connu, la distribution de Maxwell-Boltzmann permet d'estimer quantité de particules de déplaçant à une vitesse $v \pm \Delta v$. En ce sens, la distribution f est une généralisation de la densité, car elle représente la densité de particules ou de masse par unité de vitesse. Dans le même ordre d'idées, et de façon toute aussi générale, Viggen [6] fournit une dérivation de la distribution de Maxwell-Boltzmann qui représente la densité de masse volumique par unité de vitesse:

$$f(\mathbf{v}) = \rho \left(\frac{m}{2\pi k_B T} \right)^{\frac{3}{2}} e^{\left(\frac{-m(\mathbf{v} \cdot \mathbf{v})}{2k_B T} \right)}, \quad (1.2)$$

où ρ est la masse volumique, ou la densité de masse par unité de volume du gaz. Il est important de noter que dans l'Éq. (1.2), la vitesse \mathbf{v} apparaît comme une quantité vectorielle et que le terme $4\pi v^2$ de l'Éq. (1.1) n'est plus présent. Cette différence s'explique par le fait que Viggen n'exprime pas la distribution en fonction d'une vitesse radiale axisymétrique, telle que $v = |\mathbf{v}| = \sqrt{v_x^2 + v_y^2 + v_z^2}$, mais plutôt en fonction des composantes axiales de la vitesse, $\mathbf{v} = (v_x, v_y, v_z)$. Ainsi, la distribution de Maxwell-Boltzmann exprimée selon l'Éq. (1.2) permet d'estimer la masse volumique des particules se déplaçant dans l'intervalle de vitesse $[v_x \pm \Delta v_x, v_y \pm \Delta v_y, v_z \pm \Delta v_z]$. De plus, les Éqns. (1.1)-(1.2) s'appliquent à un espace tridimensionnel. En deux dimensions, le terme $4\pi v^2$ deviendrait $2\pi v$ et l'exposant $3/2$ deviendrait $2/2$.

Par ailleurs, si l'on considère un système n'étant pas à l'équilibre thermodynamique, on peut tout de même caractériser la densité du gaz par une distribution de probabilité. Cependant, cette distribution ne sera pas celle de l'Éq. (1.2) et dépendra non seulement de la vitesse mais aussi de la position et du temps :

$$f = f(\mathbf{x}, \mathbf{v}, t). \quad (1.3)$$

Boltzmann a proposé l'équation suivante pour modéliser l'évolution temporelle de cette distribution [6,7]:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + \frac{\mathbf{F}}{\rho} \cdot \nabla_{\mathbf{v}} f = \Omega(f) . \quad (1.4)$$

Il faut noter que dans la formulation présentée par Succi [7] on ne retrouve pas la masse volumique ρ car f y représente la densité de particules et non la densité de masse volumique, comme nous avons choisi de le faire ici. En somme, l'équation de Boltzmann décrit l'évolution de la distribution f dans le temps et, dans l'espace des phases, c'est-à-dire à la fois dans les dimensions spatiale (\mathbf{x}) et de vitesse (\mathbf{v}). La partie de gauche de l'Éq. (1.4) représente une dérivée totale telle que :

$$\frac{df}{dt} = \frac{\partial f}{\partial t} \frac{dt}{dt} + \frac{\partial f}{\partial \mathbf{x}} \cdot \frac{d\mathbf{x}}{dt} + \frac{\partial f}{\partial \mathbf{v}} \cdot \frac{d\mathbf{v}}{dt} , \quad (1.5)$$

ou

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + \mathbf{a} \cdot \nabla_{\mathbf{v}} f . \quad (1.6)$$

Le terme \mathbf{a} est le vecteur d'accélération des particules, égal au terme \mathbf{F}/ρ de l'Éq. (1.4) dans lequel \mathbf{F} représente une force par unité de volume. Comme une des hypothèses de la théorie cinétique des gaz stipule qu'aucune interaction, autre que les collisions, ne se produit entre les particules, le terme de force représente une force externe, telle la gravité par exemple. L'Éq. (1.6) montre que si on omet le terme de collision de l'équation de Boltzmann, cette dernière a la forme d'une équation d'advection de la distribution f , à la fois dans la dimension spatiale par le terme $\mathbf{v} \cdot \nabla f$ et dans la dimension de vitesse par le terme $\mathbf{a} \cdot \nabla_{\mathbf{v}} f$. Le terme $\Omega(f)$ peut être considéré comme un terme source découlant de la dispersion des particules lors du processus de collision. Or, nous n'avons pas défini la forme de Ω , laquelle serait nécessaire à la résolution de l'équation de Boltzmann. Nous verrons plus loin comment un modèle de collision très simple, proposé par Bhatnagar et al. [8], a contribué au passage de l'équation de Boltzmann vers la LBM.

Ce court survol de la théorie cinétique des gaz a été effectué dans le but d'introduire la distribution de Maxwell-Boltzmann et l'équation de Boltzmann, lesquelles sont des ingrédients fondamentaux du développement de la LBM. Le domaine est évidemment plus vaste et implique des dérivations qui ne sont accessibles qu'aux lecteurs très habiles avec le langage mathématique. Dans le contexte

de cette thèse, il est suggéré au lecteur avide d'approfondir ses connaissances d'étudier le théorème-H de Boltzmann. Selon ce théorème, une quantité H , analogue à l'entropie en thermodynamique, et définie telle que [7] :

$$H(t) = - \int f \ln f \, d\mathbf{v} \, d\mathbf{x} , \quad (1.7)$$

évolue de façon strictement positive $\left(\frac{dH}{dt} \geq 0 \right)$ et atteint une valeur maximale lorsque la distribution f prend la forme de la distribution de Maxwell-Boltzmann [9]. En d'autres termes, si on considère un gaz dans un système isolé, pour lequel les particules ont initialement une distribution quelconque, les collisions entre ces dernières mèneront les distributions f vers la distribution de Maxwell-Boltzmann. La quantité H augmentera jusqu'à ce que l'état d'équilibre soit atteint. Le théorème s'apparente à la deuxième loi de la thermodynamique, car l'entropie d'un système ne peut pas diminuer.

1.1.2 Automates cellulaires de gaz sur réseau

Tout comme pour la théorie cinétique des gaz, une description approfondie des automates cellulaires de gaz sur réseau (ou LGCA, pour Lattice Gas Cellular Automata) ne sera pas fournie. Les éléments essentiels à la compréhension du développement de la LBM seront plutôt couverts. Le lecteur intéressé à en apprendre davantage sur le sujet est invité à débiter ses lectures par le livre de Wolf-Gladrow [10]. Les LGCA, qui servent à simuler des écoulements de fluides, sont une variante des automates cellulaires. Cette section commence donc par une description de ces derniers.

Un automate cellulaire est constitué d'un arrangement régulier de cellules qui peuvent se trouver dans un ensemble fini d'états. Par exemple, le plus simple de ces ensembles serait de type booléen, alors que chaque cellule serait caractérisée par deux possibilités, soit vrai ou faux. L'état de chaque cellule évolue dans le temps selon des règles qui tiennent compte à la fois de l'état de ces cellules et de celui des cellules voisines. Le modèle le plus célèbre d'automate cellulaire est sans contredit le Jeu de la vie de Conway [11]. Dans cet automate, les états sont de type booléen et les cellules sont soit vivantes ou mortes. L'évolution temporelle obéit aux règles suivantes, appliquées à toutes les cellules en même temps :

- sous-population ; si une cellule vivante a moins de deux cellules voisines vivantes alors elle meurt,
- perpétuation; si une cellule vivante a deux ou trois cellules voisines qui sont aussi vivantes alors elle survit,
- surpopulation; si une cellule vivante a plus de trois cellules voisines qui sont aussi vivantes alors elle meurt,
- reproduction; si une cellule morte a trois cellules voisines vivantes alors elle devient vivante.

La Fig. 1.1 fournit un exemple de l'évolution temporelle d'un automate cellulaire qui obéit aux règles du Jeu de la vie. L'état initial est choisi arbitrairement.

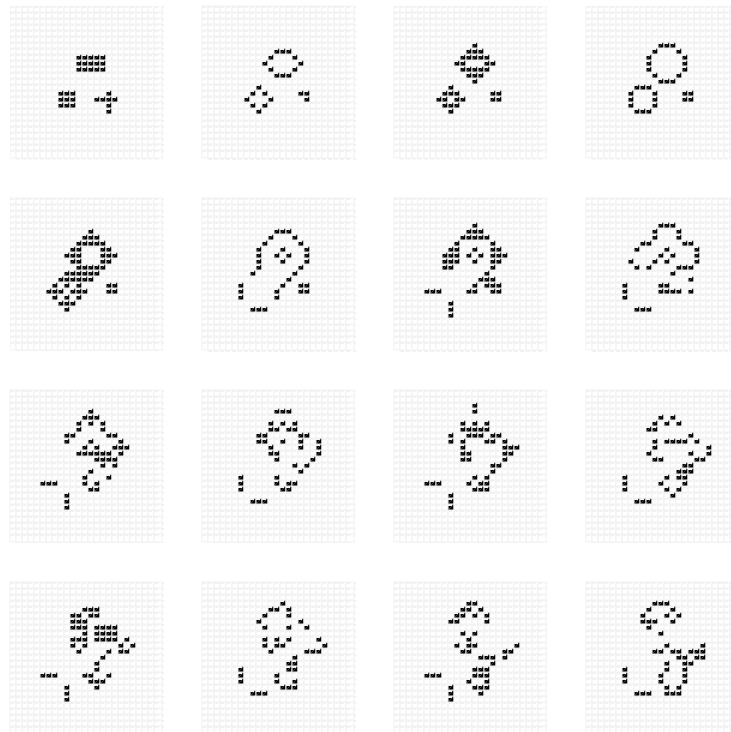


Figure 1.1 : Exemple d'évolution d'automate cellulaire selon le Jeu de la vie de Conway. L'ordre chronologique des images est de gauche à droite et de haut en bas, représentant 16 pas de temps successifs.

À priori on peut se demander à quoi servent les automates cellulaires, mis à part représenter une curiosité mathématique. Or, selon Wolfram [12], les automates, malgré leur simplicité inhérente,

permettent de créer des modèles numériques pour certains phénomènes naturels complexes tels la croissance de coquillages ou la turbulence des fluides. Dans le cadre de cette thèse, c'est l'étude des écoulements de fluides qui retient l'intérêt. Or, les LGCA, qui sont une extension des automates cellulaires, ont été créés dans cet objectif. On doit le premier modèle, communément appelé HPP, à Hardy, de Pazzis et Pomeau [13,14]. Comme pour le modèle d'automate du Jeu de la vie, le HPP utilise une grille uniforme. Cependant, il n'est pas basé sur le concept de cellules dont l'état évolue en fonction de l'état des cellules environnantes mais il s'appuie plutôt sur la représentation de chaque jonction de la grille en tant que site de collision de particules en mouvement. Le terme réseau (ou *lattice*, en anglais) est d'ailleurs utilisé pour désigner l'ensemble des sites de collision et des chemins sur lesquels les particules se déplacent. Viggen [15] résume bien l'objectif des LGCA en disant qu'il consiste à simuler de la façon la plus simple possible le comportement et les interactions du plus grand nombre possible de particules desquelles un gaz est constitué. De cette façon, les LGCA représentent en quelque sorte une méthode simple de dynamique moléculaire. Un des aspects avantageux des LGCA provient de leurs propriétés de conservation globale de masse et de quantité de mouvement, car aucune particule n'est créée ou détruite, elles sont seulement redirigées par les collisions. Ainsi, les propriétés macroscopiques comme la densité et la vitesse d'un gaz sont obtenues par un processus de moyenne spatio-temporelle à large échantillon (notre traduction libre de *coarse-graining*, en anglais) de la masse et de la vitesse de toutes les particules occupant un volume donné du réseau pendant une période donnée. La description suivante du modèle HPP permet de mieux illustrer le fonctionnement des LGCA.

Le domaine numérique du HPP est constitué de nœuds disposés de façon régulière et orthogonale. Ces nœuds sont reliés à leurs voisins des côtés est, nord, ouest et sud par des chemins rectilignes. Cet ensemble représente le réseau, illustré à la Fig. 1.2. À chaque pas de temps, les particules situées sur le même nœud entrent en collision puis sont propagées vers les nœuds voisins selon les vitesses de réseau suivantes :

$$\mathbf{c}_i = \left(c \cos\left(\frac{\pi}{2}(i-1)\right), c \sin\left(\frac{\pi}{2}(i-1)\right) \right). \quad (1.8)$$

Dans l'Éq. (1.8), i représente les quatre directions des voisins tels qu'illustrés à la Fig. 1.2 et c représente la vitesse du réseau, soit $\Delta x / \Delta t$. Dans le modèle HPP, une valeur unitaire est attribuée

à la fois à Δx et à Δt , de telle façon que $c=1$. La vitesse est donc en unité du réseau et la correspondance avec les unités réelles du problème à résoudre se fait au moyen de :

$$c^p = \left(\frac{\Delta x^p}{\Delta x} \right) \left(\frac{\Delta t}{\Delta t^p} \right) c = \frac{\Delta x^p}{\Delta t^p} c , \quad (1.9)$$

où Δx^p et Δt^p sont les équivalents de l'espacement des noeuds du réseau et du pas de temps en unités physiques.

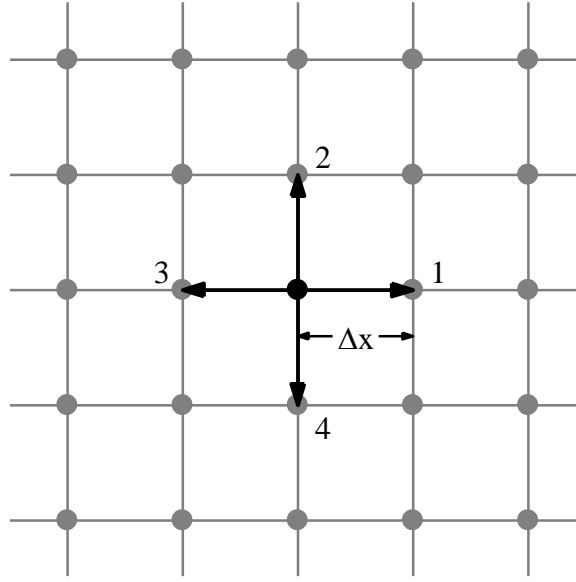


Figure 1.2 : Schéma du réseau pour le modèle HPP.

La densité de particules à chaque nœud, en unités du réseau, est obtenue au moyen de :

$$\rho(\mathbf{x}, t) = \sum_i n_i(\mathbf{x}, t) , \quad (1.10)$$

où n_i constitue une mesure booléenne, soit 0 ou 1, du nombre de particules situé sur un nœud donné et se dirigeant dans l'une des quatre directions du réseau. Chaque nœud peut donc être occupé par un maximum de quatre particules et peut aussi bien être vide. La quantité de mouvement à chaque nœud est calculée selon une approche similaire :

$$\rho(\mathbf{x}, t) \mathbf{u}(\mathbf{x}, t) = \sum_i n_i(\mathbf{x}, t) \mathbf{c}_i . \quad (1.11)$$

On peut reconstituer la densité en unités physiques en attribuant une masse appropriée à chaque particule et en divisant la densité obtenue par l'Éq. (1.10) par $(\Delta x^p)^2$. Quant à elle, la vitesse en

unités physiques est reconstituée en multipliant la vitesse \mathbf{u} de l'Éq. (1.11) par $\Delta x^p / \Delta t^p$. Évidemment, les mesures des Éq. (1.10)-(1.11) ne peuvent pas fournir des variations lisses au niveau macroscopique puisque la densité à chaque nœud ne compte que cinq valeurs possibles. Pour cette raison, on calcule des valeurs moyennes de quantité de mouvement et de densité en incluant tous les nœuds d'une région continue pour un nombre donné de pas de temps successifs. Cela permet d'obtenir une vitesse moyenne pour la région, que l'on peut ensuite convertir en unités physiques par l'Éq. (1.9). Par ailleurs, rien n'a été mentionné jusqu'ici sur le processus de collision. Il en existe seize possibilités [14], selon les particules présentes dans chaque direction. Toutes les collisions possibles conservent la quantité de mouvement. Cependant, seulement deux types de collisions modifient les mesures d'occupation (n_1, n_2, n_3, n_4) , soit les collisions frontales de deux particules pour lesquelles ces particules sont déviées à angle droit. Ainsi, la configuration $(1, 0, 1, 0)$ devient $(0, 1, 0, 1)$ et réciproquement la configuration $(0, 1, 0, 1)$ devient $(1, 0, 1, 0)$.

Pour ce qui est des conditions aux frontières, nous n'entrerons pas dans les détails dans le cas où une vitesse est imposée sur un côté du domaine. Par contre, pour une paroi solide, une particule s'y dirigeant est simplement reflétée vers le nœud d'où elle émane. Des conditions aux frontières périodiques peuvent aussi être employées pour lesquelles une particule propagée vers l'extérieur du domaine entre à nouveau dans le domaine, du côté opposé.

Un avantage indéniable de la méthode HPP provient du fait que les calculs sont toujours effectués sur des nombres entiers, donc qu'il n'y a pas d'erreurs d'arrondissements. Cependant, comme le mentionne Viggen [15], les particules n'atteignent jamais un état stationnaire et leur mouvement continu introduit du bruit statistique. Ce bruit peut être atténué en effectuant les moyennes de densité et de quantité de mouvement sur un plus grand domaine spatio-temporel mais il ne peut jamais être éliminé. De plus, Succi [7] indique que le modèle HPP permet de représenter des équations de conservation macroscopiques mais que pour la quantité de mouvement le modèle n'est pas équivalent aux équations de Navier-Stokes, car il souffre d'un manque d'invariance rotationnelle et ne peut pas reproduire le tenseur de flux de quantité de mouvement (momentum flux tensor, en anglais). Succi considère ce tenseur comme un indicateur d'isotropie. Pour des fins de simplification d'illustration du propos, il n'y inclut pas le terme dissipatif:

$$\begin{bmatrix} T_{xx} & T_{xy} \\ T_{yx} & T_{yy} \end{bmatrix} = \rho \begin{bmatrix} u_x^2 & u_x u_y \\ u_y u_x & u_y^2 \end{bmatrix} + p \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (1.12)$$

où p est la pression du fluide. Or, le tenseur obtenu pour le modèle HPP a plutôt la forme :

$$\begin{bmatrix} T_{xx} & T_{xy} \\ T_{yx} & T_{yy} \end{bmatrix} = \rho \begin{bmatrix} u_x^2 - u_y^2 & 0 \\ 0 & u_y^2 - u_x^2 \end{bmatrix} + p \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (1.13)$$

Le modèle HPP ne peut donc pas reproduire le régime hydrodynamique macroscopique. D'ailleurs, pour appuyer ce propos de façon anecdotique, Succi [7] indique avoir obtenu des vortex carrés lors de simulations. Toutefois, Frisch, Hasslacher et Pomeau [16] ont permis de pallier cette lacune que représente le manque d'invariabilité rotationnelle, en proposant un nouveau modèle LGCA, communément appelé FHP.

Le modèle FHP diffère du modèle HPP principalement selon deux aspects. Premièrement, chaque nœud a six voisins, illustrés à la Fig. 1.3, et il existe donc six vecteurs de vitesse de réseau, tels qu'indiqués par l'Éq. (1.14).

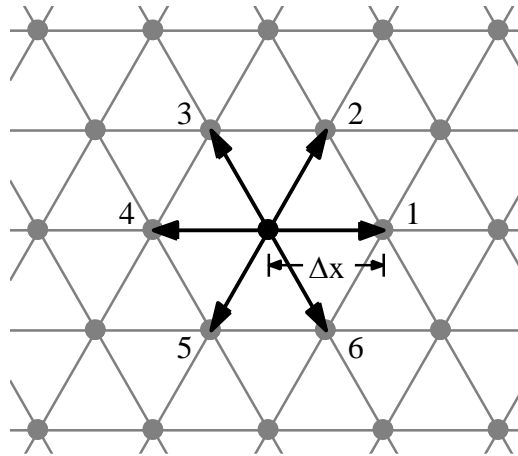


Figure 1.3 : Schéma du réseau pour le modèle FHP.

$$\mathbf{c}_i = \left(c \cos\left(\frac{\pi}{3}(i-1)\right), c \sin\left(\frac{\pi}{3}(i-1)\right) \right) \quad (1.14)$$

Deuxièmement, il existe plus de types de collisions qui changent les mesures d'occupation n_i . Elles sont détaillées dans le Tableau 1.1. Toutes les autres mesures d'occupation $(n_1, n_2, n_3, n_4, n_5, n_6)$ possibles demeurent inchangées par le processus de collision. On constate que

si trois particules réparties de façon symétrique angulairement entrent en collision, elles sont reflétées d'où elles viennent. On constate aussi que si deux particules entrent en collision face à face, il existe deux possibilités de configurations post-collisions. Ces dernières sont choisies de façon aléatoire en leur ayant au préalable attribué des probabilités égales. Comme le mentionne Viggen [15], au contraire du modèle HPP, le modèle FHP n'est pas déterministe et ne possède pas la propriété de réversibilité temporelle, c'est-à-dire qu'on ne peut pas inverser les pas de temps et retrouver la configuration initiale du réseau.

Tableau 1.1 : Types de collisions du modèle FHP.

Pré-collision	Post-collision
$(n_1, n_2, n_3, n_4, n_5, n_6)$	$(n_1, n_2, n_3, n_4, n_5, n_6)$
(1,0,0,1,0,0)	(0,1,0,0,1,0)
	(0,0,1,0,0,1)
(0,1,0,0,1,0)	(1,0,0,1,0,0)
	(0,0,1,0,0,1)
(0,0,1,0,0,1)	(1,0,0,1,0,0)
	(0,1,0,0,1,0)
(1,0,1,0,1,0)	(0,1,0,1,0,1)
(0,1,0,1,0,1)	(1,0,1,0,1,0)

L'avantage du modèle FHP, qui lui a permis de marquer un progrès dans le domaine des LGCA, consiste en sa correspondance aux équations de continuité et de Navier-Stokes dans la limite de l'incompressibilité. Chopard et al. [17] en ont d'ailleurs fait la démonstration au moyen d'un procédé d'expansion multi-échelles, en indiquant toutefois que les équations de quantité de mouvement obtenues différaient des équations standards N-S par la présence d'un coefficient $G(\rho)$ dans le terme d'advection. Un terme similaire, identifié $g(\rho)$, est aussi présent dans les équations produites par d'Humières et al. [18]. L'expansion multi-échelle a par ailleurs aussi permis d'obtenir une expression pour la viscosité cinématique, formulée de la façon suivante par Viggen [15] :

$$\nu = \frac{\Delta x^2}{\Delta t} \left(\frac{1}{2\rho_0(1-\rho_0/6)^3} - \frac{1}{8} \right), \quad (1.15)$$

où ρ_0 représente la densité moyenne du domaine en entier. De plus, la pression du fluide est calculée en fonction de la densité locale [18] :

$$p = \frac{\rho}{2} (1 - g(\rho) \mathbf{u} \cdot \mathbf{u}) , \quad (1.16)$$

où $g(\rho)$ est une fonction non-linéaire de la densité :

$$g(\rho) = \frac{\rho - 3}{\rho - 6} . \quad (1.17)$$

La correspondance du modèle FHP avec les équations de conservation macroscopiques indique que le modèle permet de simuler des écoulements dans le régime hydrodynamique. D'ailleurs, D'Humières et al. [18] ont utilisé le modèle FHP original et des variantes du modèle utilisant sept directions de propagation, en incluant une particule au repos à $\mathbf{c}_i = (0,0)$ et en modifiant les règles de collisions, pour simuler un écoulement de Poiseuille et des écoulements sur une marche inversée (backward facing step, en anglais) et autour d'une plaque plane. Ils concluent avoir obtenu des résultats correspondant bien aux résultats expérimentaux utilisés pour les fins de comparaison. Ils soulignent que l'avantage principal de la méthode réside dans sa stabilité inhérente. Par contre, ils ne commentent pas le profil de vitesse très éparé, c'est-à-dire qui inclut du bruit, obtenu pour l'écoulement de Poiseuille. Ils ne discutent pas non plus de l'effet de l'utilisation de moyennes spatio-temporelles, ce qui aurait permis de voir si le bruit peut être atténué. Wolf-Gladrow [10] a aussi utilisé le modèle FHP, afin de simuler l'écoulement autour d'un cylindre. Il ne commente pas son résultat en termes quantitatifs mais montre que de façon qualitative, la vitesse de l'écoulement obtenu est représentative du régime hydrodynamique.

Compte tenu des résultats obtenus avec le modèle FHP et ses variantes, tout laissait croire vers la fin de la décennie 80 que les LGCA constituaient une méthode prometteuse de simulation d'écoulements fluides. Or, malgré la simplicité et la stabilité des LGCA, ils n'ont pas connu le développement escompté par la suite. Succi [7] indique plusieurs problèmes ayant contribué à ralentir la progression des LGCA pour des fins d'applications pratiques, dont la présence de bruit statistique. C'est cette lacune qui a motivé les travaux de McNamara et Zanetti [19], qui constituent le point de départ de la LBM que nous utilisons aujourd'hui.

1.1.3 Automates cellulaires et équation de Boltzmann

À la section précédente, nous avons vu que les LGCA permettent de simuler des écoulements de fluides en suivant un processus répétitif de collision et propagation. Ce processus peut être exprimé de façon générale par :

$$n_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = n_i(\mathbf{x}, t) + \Delta n_i(n_1, n_2, \dots), \quad (1.18)$$

où Δn_i représente l'effet des collisions entre toutes les particules présentes sur un nœud donné. Or, les mesures d'occupation n_i sont des quantités booléennes et tel qu'il a été mentionné précédemment, un des inconvénients découlant de cette construction des LGCA réside dans le bruit statistique observé lors du calcul des propriétés macroscopiques. L'utilisation de plus grands domaines spatio-temporels pour calculer des quantités moyennes de densité et de quantité de mouvement permet d'atténuer ce bruit mais pas de l'enrayer complètement. De plus, comme le mentionnent McNamara et Zanetti [19], il faut aussi inclure dans le calcul des moyennes l'effet des conditions initiales et conséquemment effectuer plusieurs simulations avec des conditions initiales différentes. Ce processus n'est somme toute pas des plus efficaces. Afin de pallier cet inconvénient non-négligeable, McNamara et Zanetti ont proposé d'introduire l'utilisation de l'équation de Boltzmann dans les LGCA. Nous avons présenté cette équation (1.4) dans le survol de la théorie cinétique des gaz. En omettant le terme de force, l'équation de Boltzmann est constituée des mêmes ingrédients que l'Éq. (1.18) modélisant les LCGA, c'est-à-dire des termes de variation temporelle, d'advection et de collision. Cependant, dans l'équation de Boltzmann, la variable f ne représente pas une mesure finie des populations de particules mais une distribution de probabilité que des particules se déplacent à une certaine vitesse. C'est spécifiquement cet aspect que McNamara et Zanetti ont voulu exploiter, en discrétisant l'équation de Boltzmann sur le réseau des LGCA mais en conservant la nature continue des populations se propageant d'un site à l'autre. En somme, ils ont remplacé les mesures d'occupation booléennes n_i par des fonctions continues f_i pouvant prendre n'importe quelle valeur entre 0 et 1, de telle sorte que l'équation représentant la dynamique des particules prend la forme :

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + T_i(f_{j=1,2,\dots}(\mathbf{x}, t)) \quad (1.19)$$

et s'apparente à une approximation de l'équation de Boltzmann par différence finie, où le terme de collision Ω est remplacé par T_i . Ce dernier étant une expression algébrique booléenne correspondant aux règles de collisions du modèle LGCA choisi. La vitesse et la quantité de mouvement macroscopiques sont obtenues de la même façon qu'aux Éqs. (1.10)-(1.11), en y substituant bien sûr n_i par f_i . Selon cette approche, le problème du bruit statistique est éliminé parce qu'on étudie l'évolution des quantités moyennes de particules, ce qui permet d'éviter d'avoir recours aux moyennes spatio-temporelles [19]. Pour valider cette nouvelle approche, McNamara et Zanetti ont choisi le modèle LGCA FHP-III et ont entre autres effectué un test de symétrie rotationnelle de propagation d'onde de son. En introduisant une perturbation au centre d'un domaine, c'est-à-dire en attribuant une densité plus élevée au nœud du milieu, ils ont montré que la variation spatiale de densité résultant de cette discontinuité était axisymétrique, ce qui représente une condition nécessaire afin que le modèle soit considéré valide. De plus, aucune moyenne spatiale des distributions f_i n'a été nécessaire, confirmant l'élimination du bruit statistique. On peut donc considérer l'approche de McNamara et Zanetti comme le point de départ du développement de la LBM, même si l'opérateur de collision est basé sur les règles du modèle LGCA sous-jacent choisi. Les éléments de base sont présents : 1) une équation de collision et propagation pour des particules se déplaçant sur un réseau fini, 2) un opérateur de collision et 3) des distributions de particules.

En partant aussi de la constatation que l'une des difficultés principales des LGCA provenait des fluctuations des quantités mesurées, lesquelles sont impossibles à éliminer à moins de pouvoir utiliser un nombre très élevé de nœuds sur le réseau, Higuera et Jiménez [20] ont aussi proposé l'approche de l'équation de Boltzmann pour la simulation de gaz sur réseau. Cependant, contrairement à McNamara et Zanetti, l'opérateur de collision n'est pas constitué d'opérations algébriques visant à reproduire les règles de collisions du modèle LGCA choisi, mais est plutôt un opérateur matriciel linéarisé de ces règles de collisions. En dérogeant quelque peu de la notation utilisée dans l'article original [20], l'opérateur de collision Ω est exprimé par :

$$\Omega(f_1, f_2, \dots) = \Omega(\mathbf{f}) = \mathbf{\Omega}^{(1)} \cdot (\mathbf{f} - \mathbf{f}^{eq,0} - \mathbf{f}^{eq,2}) . \quad (1.20)$$

Dans l'Éq. (1.20), \mathbf{f} est le vecteur des distributions de particules, les termes \mathbf{f}^{eq} sont les vecteurs de distributions de particules à l'équilibre et $\mathbf{\Omega}^{(1)}$ est la matrice de collision. Cette dernière est

dérivée à partir des règles de collisions d'un modèle LGCA choisi mais nous ne décrivons pas comment elle est construite. Nous ne décrivons pas non plus les expressions des distributions à l'équilibre afin de ne pas alourdir la lecture. Ce qu'il est important de constater, c'est que le modèle de collision de Higuera et Jiménez est en fait une combinaison linéaire de l'écart entre la valeur des distributions et la valeur de ces mêmes distributions à l'état d'équilibre, lequel dépend de la densité et de la quantité de mouvement du fluide. De plus, pour les variantes du modèle LGCA FHP, la matrice $\Omega^{(1)}$ ne contient que deux ou trois paramètres indépendants [20]. Dans les modèles LGCA, la viscosité du gaz dépend des règles de collisions et modifier cette viscosité revient à manipuler les règles. Or, comme l'opérateur linéarisé de collision de Higuera et Jiménez ne dépend au final que de quelques paramètres indépendants d'une matrice, ces chercheurs indiquent que la matrice $\Omega^{(1)}$ peut être déterminée d'une façon générale, découplée des règles de collisions d'un modèle spécifique LGCA. Cela permet de contrôler la viscosité sans avoir recours à un modèle LGCA sous-jacent. Cette constatation marque la séparation entre le développement de la LBM et les LGCA. Cela étant dit, pour valider leur méthode, Higuera et Jiménez ont tout de même dérivé la matrice de collision à partir des règles du modèle FHP-III. Pour la simulation d'un écoulement autour d'un cylindre à des nombres de Reynolds variant de 10 à 80, ils ont obtenu des écoulements stationnaires et instationnaires comparables à des résultats expérimentaux et à des résultats provenant de programmes basés sur la résolution des équations de Navier-Stokes en différences finies.

L'utilisation de distributions à l'équilibre dans le processus de collision a été reprise par Koelman [21]. Selon sa méthode, les distributions prennent la valeur de distributions à l'équilibre basées sur la distribution de Maxwell-Boltzmann. Elles dépendent de la densité et de la quantité de mouvement macroscopiques. L'opérateur de collision est donc linéaire, car il peut être exprimé de la façon générale suivante :

$$f'_i(\mathbf{x}, t) = 1 \cdot f_i^{eq}(\mathbf{x}, t) + 0 \cdot f_i(\mathbf{x}, t) , \quad (1.21)$$

où f_i^{eq} et f'_i représentent respectivement les distributions à l'équilibre et les distributions post-collision. Il diffère donc de l'opérateur proposé par Higuera et Jiménez par l'absence d'une matrice de collision reliant explicitement chaque f'_i à l'ensemble des f_i pré-collision (f_1, f_2, \dots) . La collision s'effectue indépendamment sur chaque direction du réseau. En réalité, ce n'est pas tout à

fait le cas, car les distributions de toutes les directions interviennent dans le calcul des distributions à l'équilibre [21]:

$$f_i^{eq}(\rho, \mathbf{j}) = \frac{W_i}{\rho_0} \left(\rho + \frac{m}{k_B T} (\mathbf{c}_i \cdot \mathbf{j}) + \frac{1}{2\rho_0} \left(\frac{m}{k_B T} \right)^2 (\mathbf{c}_i \cdot \mathbf{j})^2 - \frac{1}{2\rho_0} \left(\frac{m}{k_B T} \right)^2 (\mathbf{j} \cdot \mathbf{j})^2 \right), \quad (1.22)$$

puisque $\rho = \sum_i f_i(\mathbf{x}, t)$ et $\mathbf{j}(\mathbf{x}, t) = \rho(\mathbf{x}, t) \mathbf{u}(\mathbf{x}, t) = \sum_i \mathbf{c}_i f_i(\mathbf{x}, t)$. L'Éq. (1.22) constitue une expansion du 2^e ordre de la distribution de Maxwell-Boltzmann autour de la vitesse macroscopique \mathbf{u} , appliquée au nombre fini de vecteurs de vitesse du réseau. Cette équation inclut aussi des poids W_i permettant d'assurer une quadrature équivalente à l'intégration de la distribution continue de Maxwell-Boltzmann. Nous reviendrons plus loin sur cette quadrature, car elle est fondamentale à la validité de la LBM en tant que méthode de simulation d'écoulements fluides. On constate que la formulation de Koelman n'utilise pas de règles de collisions provenant des LGCA, c'est pourquoi le titre de « méthode de Boltzmann sur réseau » est désormais plus approprié. Par ailleurs, Koelman a dérivé le modèle pour une forme généralisée de réseau régulier, c'est-à-dire pour qu'il soit applicable tant à un réseau pleinement orthogonal qu'à un réseau de type FHP tel que présenté à la Fig. 1.3. En effet, seuls les poids W_i sont modifiés selon la forme du réseau.

Avec l'opérateur de collision tel que défini par l'Éq. (1.21), la viscosité cinématique du fluide, en unités de réseau, n'est pas un paramètre que l'on contrôle. En effet, Koelman a effectué une simulation d'atténuation d'onde de cisaillement (shear-wave damping, en anglais) et a constaté que cette viscosité est équivalente à $\frac{\Delta t}{2} \frac{k_B T}{m}$. De plus, une simulation de propagation d'onde de son a permis de déterminer que la vitesse du son c_s est équivalente à $\sqrt{k_B T / m}$, toujours pour l'opérateur de collision de l'Éq. (1.21). Or, Koelman mentionne que la viscosité cinématique peut être contrôlée en introduisant un coefficient de relaxation dans le modèle linéaire de collision :

$$f_i'(\mathbf{x}, t) = (1 + \omega) \cdot f_i^{eq}(\mathbf{x}, t) - \omega \cdot f_i(\mathbf{x}, t). \quad (1.23)$$

Attribuer une valeur positive à ω représente une sur-relaxation et permet de réduire la viscosité, ou d'augmenter le nombre de Reynolds de la simulation. Koelman ne fournit pas par contre l'expression algébrique reliant ν à ω . Il a néanmoins validé la méthode proposée au moyen de l'opérateur de collision sans relaxation. Par la simulation de propagation d'onde de son, il a

confirmé l'isotropie spatiale du modèle. Il a aussi simulé l'écoulement autour d'une plaque plane, en régimes stationnaire et instationnaire, et obtenu des résultats comparables à la littérature.

En parallèle des travaux de Koelman, Qian et al. [22] ont aussi proposé, de façon indépendante, un opérateur de collision linéaire à un seul coefficient. De plus, comme l'avait suggéré Koelman, ils ont introduit l'utilisation d'un coefficient de relaxation. Leur utilisation de ce coefficient, aussi symbolisé par ω , diffère légèrement de l'Éq. (1.23) mais le concept est identique. Ils expriment le processus complet de collision et propagation par :

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = (1 - \omega) \cdot f_i(\mathbf{x}, t) + \omega \cdot f_i^{eq}(\mathbf{x}, t). \quad (1.24)$$

De plus, pour les distributions à l'équilibre, ils utilisent aussi une expansion de la distribution de Maxwell-Boltzmann:

$$f_i^{eq} = w_i \rho \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{1}{2} \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{c_s^4} - \frac{1}{2} \frac{\mathbf{u} \cdot \mathbf{u}}{c_s^2} \right). \quad (1.25)$$

Dans l'éq. (1.25), nous n'avons pas utilisé les mêmes symboles que Qian et al. [22] afin de conserver une homogénéité avec les modèles présentés précédemment. Le lecteur notera aussi qu'en raison de la normalisation par la densité nominale ρ_0 , les poids W_i de l'Éq. (1.22) ont une unité de masse, alors que les poids w_i de l'Éq. (1.25) sont des scalaires sans unités. Cependant, pour le réseau orthogonal à neuf vitesses, dénoté d2q9 par Qian et al., la valeur des poids est définie de façon similaire pour les deux méthodes, tel que montré au Tableau 1.2. Dans ce tableau, les vitesses sont exprimées en unités unitaires de réseau, en utilisant $\Delta t = 1$ et $\Delta x = 1$.

Tableau 1.2 : Valeur des poids pour les distributions à l'équilibre.

Direction	$c_{i,x}$	$c_{i,y}$	W_i/ρ_0 ou w_i
1	0	0	4/9
2	1	0	1/9
3	0	1	1/9
4	-1	0	1/9
5	0	-1	1/9
6	1	1	1/36
7	-1	1	1/36
8	-1	-1	1/36
9	1	-1	1/36

Comme il a été mentionné précédemment, le choix des poids n'est pas aléatoire. Qian et al. indiquent que ces poids sont entre autres déterminés dans le but d'obtenir l'invariance galiléenne du modèle. Comme on retrouve ce dernier concept très fréquemment dans la littérature LBM, nous y reviendrons plus en détail. Par ailleurs, Qian et al. indiquent aussi qu'en utilisant une technique d'expansion multi-échelles, on peut montrer que le modèle représenté par les Éqns. (1.24)-(1.25) est équivalent aux équations de Navier-Stokes et que la vitesse du son et la viscosité cinématique sont fournies par :

$$c_s = \frac{1}{\sqrt{3}} , \quad (1.26)$$

$$\nu = \frac{1}{6} \left(\frac{2}{\omega} - 1 \right) . \quad (1.27)$$

On constate que le coefficient de relaxation ω permet d'ajuster la viscosité, ce qui confirme ce que Koelman [21] avait avancé. Qian et al. ont confirmé la validité des Éqns. (1.26)-(1.27) par la méthode de perturbation en simulant numériquement l'évolution d'une onde sinusoïdale et en mesurant la vitesse du son et la viscosité.

En résumé, dans cette section nous avons montré que le passage des LGCA vers la LBM s'est fait par trois éléments-clés :

- 1) Introduction de l'équation de Boltzmann, donc de l'utilisation de distributions continues de particules plutôt que de représentations booléennes de celles-ci,

- 2) Linéarisation du processus de collision et abandon des règles de collisions issues des LGCA,
- 3) Utilisation de distributions à l'équilibre basées sur la distribution de Maxwell-Boltzmann.

Dans la littérature, on réfère à la méthode LBM présentée par Qian et al. en utilisant l'acronyme LBM BGK, ou LBGK, parce que Bhatnagar, Gross et Krook [8] furent les premiers à proposer l'utilisation d'un opérateur de collision linéaire basé sur une relaxation dans l'équation de Boltzmann. Nous allons maintenant décrire de façon plus formelle les équations et la marche à suivre permettant de simuler un écoulement fluide avec la LBM.

1.1.4 Méthode de Boltzmann sur réseau

La LBM constitue une méthode de simulation numérique d'écoulements de fluides qui traite de ces derniers d'un point de vue mésoscopique, plutôt que de les considérer comme un milieu continu tel qu'on le fait pour construire des résolveurs basés sur les équations de Navier-Stokes. Le fluide est perçu comme un ensemble probabiliste de particules qui interagissent et pour lequel on reconstruit les propriétés macroscopiques du fluide en calculant les moments des distributions de particules par les vecteurs de vitesses définissant le réseau.

Dans le présent travail toutes les simulations seront exécutées en deux dimensions et le type de réseau choisi est le d2q9, mentionné à la section précédente et illustré à la Fig. 1.4.

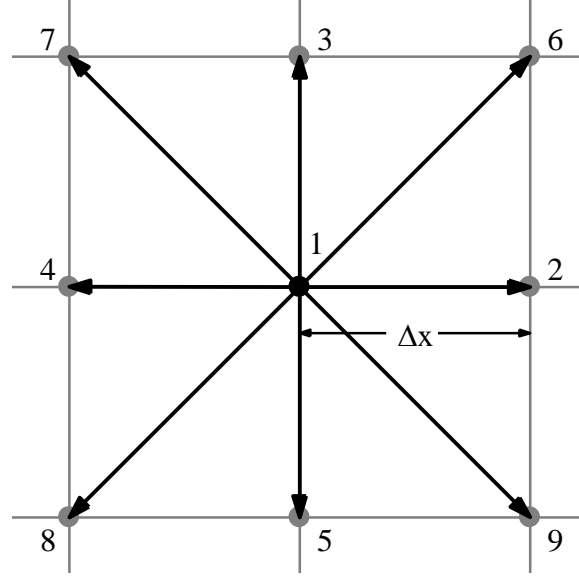


Figure 1.4 : Schéma du réseau d2q9.

Ce réseau comporte neuf vitesses, exprimées par :

$$\mathbf{c}_i = (c_x, c_y)_i = \begin{cases} (0,0) & i = 1 \\ c \left(\cos \left[\pi(i-2)/2 \right], \sin \left[\pi(i-2)/2 \right] \right) & i = 2, 3, 4, 5 \\ c \left(\cos \left[\pi(i-6)/2 + \pi/4 \right], \sin \left[\pi(i-6)/2 + \pi/4 \right] \right) & i = 6, 7, 8, 9 \end{cases} \quad (1.28)$$

où $c = \Delta x / \Delta t$ est la vitesse générale du réseau. Dans les modèles présentés à la section 1.1.3, il était assumé que $\Delta t = \Delta x = 1$. Or, pour le présent travail, cette approche ne sera pas utilisée. La distance réelle Δx entre les nœuds du réseau sera utilisée telle quelle et le pas de temps Δt sera ajustable, bien qu'en pratique il sera égalisé à Δx afin d'utiliser une vitesse unitaire du réseau $c = 1$. La correspondance entre les unités physiques et les unités du réseau se fera par le nombre de Reynolds et sera expliquée en détail plus loin.

La LBM consiste à simuler l'écoulement d'un fluide en résolvant l'Éq. (1.24), représentée ici avec une notation différente pour le coefficient de relaxation :

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + \frac{\Delta t}{\Delta t_R} (f_i^{eq}(\mathbf{x}, t) - f_i(\mathbf{x}, t)) . \quad (1.29)$$

Cette forme a été choisie afin de montrer que l'équation de base de la LBM est en fait la combinaison d'une approximation en différence finie de l'équation de Boltzmann en version discrète, soit

$$\left[\frac{\partial f_i(\mathbf{x}, t)}{\partial t} + \mathbf{c}_i \cdot \nabla f_i(\mathbf{x}, t) = \Omega_i \right] \approx \left[\frac{f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t)}{\Delta t} = \Omega_i \right], \quad (1.30)$$

et de l'opérateur de collision tel que proposé par Bhatnagar, Gross et Krook [8] :

$$\Omega_i = \frac{1}{\Delta t_R} (f_i^{eq}(\mathbf{x}, t) - f_i(\mathbf{x}, t)). \quad (1.31)$$

La variable Δt_R représente le temps de collision. Ainsi, le coefficient de relaxation $\Delta t / \Delta t_R$ est le rapport du pas de temps et du temps de collision. L'Éq. (1.29) se résout en deux étapes. Premièrement, sur chaque nœud du réseau, on calcule des distributions intermédiaires issues du processus de collision :

$$f_i'(\mathbf{x}, t) = f_i(\mathbf{x}, t) + \frac{\Delta t}{\Delta t_R} (f_i^{eq}(\mathbf{x}, t) - f_i(\mathbf{x}, t)). \quad (1.32)$$

Ensuite, on propage ces distributions sur les nœuds pointés par les vecteurs de vitesse :

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i'(\mathbf{x}, t). \quad (1.33)$$

On calcule les propriétés macroscopiques au moyen de moments des distributions dans l'espace fini des vitesses :

$$\rho(\mathbf{x}, t) = \sum_i f_i(\mathbf{x}, t), \quad (1.34)$$

$$\mathbf{u}(\mathbf{x}, t) = \frac{\sum_i \mathbf{c}_i f_i(\mathbf{x}, t)}{\rho(\mathbf{x}, t)}. \quad (1.35)$$

Ces quantités servent au calcul des distributions à l'équilibre présentes dans l'opérateur de collision, ce qui permet de commencer un autre cycle de collision-et-propagation. Les distributions à l'équilibre proviennent d'une expansion en série de Taylor au 2^e ordre de la distribution de Maxwell-Boltzmann, centrée autour de la vitesse macroscopique nulle. En utilisant la relation d'équivalence $c_s^2 = k_B T / m$, identifiée entre autres par Koelman [21], la distribution de Maxwell-

Boltzmann en deux dimensions centrée autour d'une vitesse macroscopique arbitraire est exprimée par :

$$f(\mathbf{c}, \rho, \mathbf{u}) = \rho \left(\frac{1}{2\pi c_s^2} \right) e^{\left(\frac{-(\mathbf{c}-\mathbf{u})(\mathbf{c}-\mathbf{u})}{2c_s^2} \right)}. \quad (1.36)$$

La notation \mathbf{v} pour le vecteur de vitesse des particules, telle qu'utilisée dans l'Éqn. (1.2), est remplacée par $\mathbf{v} = \mathbf{c} - \mathbf{u}$ afin d'introduire le symbole de vitesse du réseau et de montrer que la distribution a subi une translation par la vitesse macroscopique non-nulle. L'expansion en série de Taylor de l'Éqn. (1.36) autour de $\mathbf{u} = (0,0)$ est :

$$f(\mathbf{c}, \rho, \mathbf{u}) \approx \frac{\rho}{2\pi c_s^2} \exp\left(\frac{-\mathbf{c} \cdot \mathbf{c}}{2c_s^2}\right) \left(1 + \frac{\mathbf{c} \cdot \mathbf{u}}{c_s^2} + \frac{1}{2} \frac{(\mathbf{c} \cdot \mathbf{u})^2}{c_s^4} - \frac{1}{2} \frac{\mathbf{u} \cdot \mathbf{u}}{c_s^2} \right). \quad (1.37)$$

Afin d'obtenir les distributions à l'équilibre pour les vitesses du réseau, on exprime l'Éqn. (1.37) de la façon suivante :

$$f_i^{eq}(\rho, \mathbf{u}) \approx \rho w_i \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{1}{2} \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{c_s^4} - \frac{1}{2} \frac{\mathbf{u} \cdot \mathbf{u}}{c_s^2} \right). \quad (1.38)$$

Les poids w_i et la constante c_s doivent être choisis afin de remplir les conditions suivantes, pour lesquelles la notation a été simplifiée par souci d'allègement :

$$\rho = \sum_i f_i^{eq}, \quad (1.39)$$

$$\mathbf{u} = \frac{\sum_i \mathbf{c}_i f_i^{eq}}{\rho}. \quad (1.40)$$

Il est impératif que les moments hydrodynamiques des Éqns. (1.39)-(1.40) obtenus au moyen de la quadrature de l'Éqn. (1.38) constituent des approximations les plus fidèles possibles de l'intégration des moments de l'Éqn. (1.36) dans l'espace continu des vitesses microscopiques. Pour ce faire, les poids w_i et la constante c_s sont obtenus au moyen d'une quadrature de type Gauss-Hermite [23]. De plus, tel qu'indiqué par Lätt [24], les vitesses du réseau, la constante c_s et les poids doivent remplir une série de conditions de symétrie afin que la LBM permette de résoudre

correctement un écoulement de fluide. Ces conditions sont présentées en conservant la notation d'Einstein utilisée par Lätt car elle permet d'exprimer les équations de façon compacte. En deux dimensions, les indices grecs $(\alpha, \beta, \gamma, \lambda, \varepsilon)$ représentent l'axe x ou l'axe y .

$$\begin{aligned}
 \sum w_i &= 1 \\
 \sum w_i c_{i,\alpha} &= 0 \\
 \sum w_i c_{i,\alpha} c_{i,\beta} &= c_s^2 \delta_{\alpha,\beta} \\
 \sum w_i c_{i,\alpha} c_{i,\beta} c_{i,\gamma} &= 0 \\
 \sum w_i c_{i,\alpha} c_{i,\beta} c_{i,\gamma} c_{i,\lambda} &= c_s^4 (\delta_{\alpha,\beta} \delta_{\gamma,\lambda} + \delta_{\alpha,\gamma} \delta_{\beta,\lambda} + \delta_{\alpha,\lambda} \delta_{\beta,\gamma}) \\
 \sum w_i c_{i,\alpha} c_{i,\beta} c_{i,\gamma} c_{i,\lambda} c_{i,\varepsilon} &= 0
 \end{aligned} \tag{1.41}$$

Les poids et la vitesse spécifiés par He et Luo [23] et Qian et al. [22], soit $\mathbf{w} = (16, 4, 4, 4, 4, 1, 1, 1, 1)/36$ et $c_s = 1/\sqrt{3}$, remplissent ces conditions de symétrie. De plus, Qian et al. affirment qu'ils permettent d'obtenir l'invariance galiléenne.

Le concept d'invariance galiléenne est mentionné fréquemment dans la littérature LBM. De façon générale, il stipule que les équations de mouvement doivent être les mêmes, indépendamment du cadre de référence inertiel. C'est-à-dire que les lois de Newton s'appliquent à tous les référentiels qui résultent d'une transformation inertielle d'un autre référentiel. Ainsi, l'accélération d'un corps ou d'une particule dans un référentiel fixe est égale à l'accélération considérée dans un référentiel se déplaçant à une vitesse constante. Dans le cadre de la LBM, cela implique par exemple que pour un fluide au repos dans lequel on introduit une perturbation, son champ de vitesse calculé par des moments de distributions doit être identique à celui d'un écoulement à vitesse constante auquel on introduit la même perturbation et pour lequel on soustrait cette constante des moments des distributions. Le concept d'invariance galiléenne est important dans l'expression des distributions à l'équilibre car elles sont définies en fonction d'une infinité de référentiels inertiels par la vitesse locale du fluide \mathbf{u} . Le processus de collision de type BGK, basé sur une relaxation vers des distributions à l'équilibre, doit permettre d'obtenir l'invariance galiléenne ou du moins un niveau d'invariance galiléenne suffisant.

Dans la littérature LBM, ce concept est aussi utilisé pour désigner l'invariance rotationnelle, c'est-à-dire que l'écoulement résolu doit être le même, indépendamment de l'orientation du réseau. Les conditions de symétrie de l'Éqn. (1.41) servent à produire ce type d'invariance.

Par ailleurs, une des lacunes identifiées par Succi [7] concernant les LGCA était justement le manque d'invariance galiléenne. Pour le modèle FHP, cette lacune se traduit par l'introduction du coefficient $g(\rho)$ dans le terme d'advection des équations macroscopiques équivalentes [18]. L'éqn. (1.16) montre aussi que par ce coefficient l'expression de la pression du modèle FHP dépend de la vitesse, ce qui contredit le concept d'invariance galiléenne.

Il est possible de démontrer que le modèle LBM basé sur l'opérateur de collision BGK et le réseau d2q9 est équivalent à résoudre les équations de Navier-Stokes en régime faiblement compressible. Cette démonstration a été faite au moyen d'un procédé d'expansion multi-échelles Chapman-Enskog par plusieurs auteurs tels Chen et Doolen [25], Lätt [24] et Viggen [15]. Cette démonstration est aussi fournie dans le présent travail, que le lecteur est invité à aller consulter à l'annexe A. De cette dérivation découlent quatre relations constitutives importantes de la LBM, lesquelles permettent, entre autres, d'établir la correspondance avec la dynamique macroscopique de l'écoulement de fluide simulé :

$$\nu = c_s^2 \Delta t \left(\frac{\Delta t_R}{\Delta t} - \frac{1}{2} \right), \quad (1.42)$$

$$p = \rho c_s^2, \quad (1.43)$$

$$\Pi = \sum_{i=1}^N (\mathbf{c}_i \mathbf{c}_i^T) \left(f_i^{eq} + \left(1 - \frac{1}{2} \frac{\Delta t}{\Delta t_R} \right) f_i^{neq} \right), \quad (1.44)$$

$$f_i^{neq} = -\Delta t_R \left(\mathbf{c}_i \cdot \nabla + \frac{\partial}{\partial t} \right) f_i^{eq}. \quad (1.45)$$

Notons que nous utilisons le terme *fluide* plutôt que *gaz* pour désigner la phase de la matière pour laquelle nous désirons simuler la dynamique au moyen de la LBM. En effet, bien que la LBM avec opérateur de collision de type BGK ait été conçue à partir de la théorie cinétique des gaz et des LGCA, pour lesquels on assumait que le volume occupé par toutes les particules était négligeable en comparaison du volume occupé par le gaz, la dérivation des équations de Navier-Stokes nous montre que les conditions nécessaires à ce que la LBM corresponde à ces équations sont : 1) l'écoulement doit être faiblement compressible et 2) le fluide doit être de type newtonien. Donc, la méthode est applicable à tous les fluides qui remplissent le deuxième critère, comme l'air et l'eau. De plus, en raison des hypothèses sous-jacentes à la théorie cinétique des gaz, cette dernière assume

que seules des collisions de deux particules se produisent, car il est improbable que plusieurs particules entrent en collision en même temps. Or, étant donné que pour un liquide l'espace occupé par les particules est nécessairement plus grand que pour un gaz, on pourrait anticiper que des collisions à plusieurs particules soient plus probables et que la théorie sur laquelle la LBM est fondée ne soit pas applicable. En réalité la LBM peut aussi être appliquée aux liquides. Dans l'expansion Chapman-Enskog le terme ε correspond au nombre de Knudsen [24], qui représente le rapport de la distance moyenne parcourue par les particules entre chaque collision à la longueur caractéristique de l'écoulement simulé, $Kn = \lambda/L$. De plus, une condition nécessaire à ce que la LBM corresponde aux équations de Navier-Stokes et permette de résoudre des écoulements dans le régime hydrodynamique est que le nombre de Knudsen soit faible [24]. Donc le type de fluide, gaz ou liquide, n'est pas vraiment important dans la mesure où le domaine spatial de la simulation est suffisamment grand.

1.1.4.1 Unités du réseau et unités physiques

Afin de compléter la description de la LBM, il faut préciser comment faire le passage des unités employées dans le processus de collision-et-propagation des distributions vers les unités physiques du problème à résoudre. Cette équivalence se fait au moyen du nombre de Reynolds de la simulation, tel que montré à l'Éqn. (1.46). Dans cette équation et les équations subséquentes, les exposants P et B désignent respectivement les unités physiques et du réseau.

$$\text{Re} = \frac{\rho^P U_\infty^P L^P}{\mu^P} = \frac{\rho^B U_\infty^B L^B}{\mu^B} \quad (1.46)$$

Il n'existe pas qu'une seule façon de choisir les valeurs de ρ^B , U_∞^B et L^B . Pour le présent travail nous choisissons de ne pas modifier les unités de distance, c'est-à-dire que $L^B = L^P$. Donc, l'espace entre chaque nœud du réseau est $\Delta x^B = \Delta x^P$. Comme nous employons une vitesse du réseau unitaire, $c = 1 = \Delta x^B / \Delta t^B$, nous assignons au pas de temps Δt^B la même valeur que Δx^B . Par contre, la valeur du pas de temps réel, Δt^P , n'est quant à elle pas égale à celle de la distance Δx^P . On la calcule plutôt au moyen d'une équivalence de distance parcourue normalisée :

$$\frac{U_\infty^B \Delta t^B}{\Delta x^B} = \frac{U_\infty^P \Delta t^P}{\Delta x^P},$$

$$\Delta t^P = \Delta t^B \frac{U_\infty^B}{U_\infty^P} \frac{\Delta x^P}{\Delta x^B} . \quad (1.47)$$

Comme dans le cas présent il a été choisi d'utiliser $\Delta x^B = \Delta x^P$, le pas de temps réel se calcule simplement par $\Delta t^P = \Delta t^B U_\infty^B / U_\infty^P$. Il ne reste donc qu'à définir U_∞^B . Puisque la dérivation des équations de Navier-Stokes présentée à l'annexe A s'appuie sur la condition que l'écoulement soit faiblement compressible, ce qui nécessite que $|\mathbf{u}| \ll c_s$, la vitesse de référence U_∞^B doit aussi être largement inférieure à la vitesse du son du réseau. Pour le présent travail, nous avons choisi d'utiliser la valeur suivante : $U_\infty^B = 0.1c_s$. Le choix de cette dernière est un compromis entre le respect du critère de bas nombre de Mach du réseau et le désir de ne pas obtenir un pas de temps réel trop petit, car cela allonge la période de convergence des simulations vers un état stationnaire.

1.1.5 Évolution de la méthode de Boltzmann sur réseau

Dans la section précédente, la LBM avec opérateur de collision de type BGK a été détaillée. Cette méthode constitue en quelque sorte la base de la LBM appliquée à la résolution d'écoulements fluides. Or, pour appliquer la méthode à des cas spécifiques, il faut aussi considérer d'autres aspects, tels l'application de conditions aux frontières et le raffinement de maillage. De plus, au cours des vingt-cinq dernières années, la méthode a beaucoup évolué et diverses branches de recherche ont émergé, permettant de résoudre des écoulements dans des conditions variées. Le cadre théorique de la thèse sera donc complété en dressant un bref portrait de diverses méthodes développées dans le cadre de la LBM qui sont pertinentes au présent travail. Par contre, les aspects théoriques du développement des méthodes présentées ne seront pas approfondis ici.

1.1.5.1 Conditions aux frontières

Tout d'abord, la simulation numérique d'un écoulement fluide nécessite qu'on définisse les frontières du domaine numérique associé, sur lesquels on impose des conditions qui agissent comme force motrice de l'écoulement. Dans le contexte de la LBM pour les écoulements faiblement compressibles, il faut imposer la vitesse et la pression, ou la densité, sur les nœuds du réseau qui forment le périmètre du domaine et aussi sur les nœuds se trouvant à proximité d'un objet placé à l'intérieur du domaine. Pour ce faire, les quantités macroscopiques imposées doivent être traduites en distributions f_i , car à la suite du processus de propagation certaines distributions

sont forcément manquantes. Beaucoup de méthodes ont été développées pour accomplir cette tâche. Par exemple, si l'on considère une paroi solide alignée avec le réseau et située à une distance de $0.5\Delta x$ de ce dernier, pour laquelle on désire imposer une condition de glissement nul, la méthode la plus simple et commune consiste à faire réfléchir les distributions de particules vers le nœud d'où elles proviennent. Pour une paroi située au bas du domaine, en se référant à la Fig. 1.4, cela implique que $f_3 = f'_5$, $f_6 = f'_8$, et $f_7 = f'_9$. Cette méthode de réflexion peut aussi être utilisée pour imposer une condition de glissement, en utilisant les distributions des nœuds voisins: $f_3(x, y) = f'_5(x, y)$, $f_6(x, y) = f'_9(x - \Delta x, y)$ et $f_7(x, y) = f'_8(x + \Delta x, y)$. Par contre, cette méthode ne permet pas d'imposer une vitesse donnée. Les méthodes de Zou et He [26], Bouzidi et al. [27] et Guo et al. [28], pour ne nommer que celles-là, pallient cette lacune. La première positionne les nœuds sur la paroi et est basée sur la réflexion de la partie non-équilibrée des distributions, c'est-à-dire que les distributions manquantes sont reconstruites en combinant les distributions non-équilibrées réfléchies et des distributions à l'équilibre calculées avec la vitesse macroscopique imposée. Cette méthode peut aussi être utilisée pour imposer une pression donnée aux limites du domaine. La méthode de Bouzidi et al. est basée sur une généralisation de la réflexion des particules, pour une distance arbitraire entre le nœud et la paroi. On y assume que la distribution réfléchie sur un nœud a parcouru une distance totale égale à Δx , ce qui nécessite une interpolation au point d'émanation de la distribution qui frappe la paroi. Par exemple, pour une paroi située du côté droit à $q\Delta x$, où q est inférieur à 0.5, on aurait $f_4(x, y) = 2q f'_2(x, y) + (1 - 2q) f'_2(x - \Delta x, y)$. Afin d'imposer une vitesse sur la paroi, une correction est imposée, telle que $\delta f_i \propto \alpha_i (\mathbf{c}_i \cdot \mathbf{U}_w)$, où \mathbf{U}_w représente la vitesse imposée. Cette méthode peut donc être imposée sur la paroi courbe d'un objet pour laquelle la distance avec les nœuds du réseau varie. La méthode de Guo et al. le peut aussi, car elle assume que les distributions manquantes sur les nœuds près des parois sont propagées à partir de nœuds fictifs situés à l'extérieur du domaine ou à l'intérieur d'un objet, comme si le réseau était prolongée d'une couche de nœuds. Ces distributions sont issues d'un processus de collision basé sur les parties à l'équilibre et non-équilibrées de distributions reconstruites à partir d'extrapolations provenant des nœuds fluides. La vitesse imposée sur la paroi sert à l'extrapolation de la vitesse utilisée pour construire les distributions à l'équilibre. Pour une paroi située à la droite d'un nœud, le résultats de cette démarche est illustré par $f_4(x, y, t + \Delta t) = \bar{f}_4^{eq}(x + \Delta x, y, t) + (1 - \Delta t / \Delta t_R) \bar{f}_4^{neq}(x + \Delta x, y, t)$, où la

notation \bar{f} indique les quantités provenant d'extrapolation qui dépendent de la distance entre le nœud fluide et la paroi. Ces méthodes décrites brièvement montrent que le réseau régulier de la LBM n'empêche pas de traiter de conditions aux frontières sur surfaces courbes. Une autre approche a aussi été adaptée au contexte de la LBM pour traiter de ce type de conditions, soit l'emploi de conditions aux frontières immergées.

Les conditions aux frontières immergées s'appliquent à des objets que l'on place sur le réseau sans invalider les nœuds se trouvant à l'intérieur de l'objet. C'est-à-dire que le processus de collision-et-propagation est appliqué à tous les nœuds et que des corrections sont appliquées aux distributions des nœuds voisins de la paroi de l'objet pour tenir compte de l'impact du mouvement de cette paroi. Par exemple, Feng et Michaelides [29] ont introduit un terme de force dans l'équation de collision de la LBM permettant aux nœuds du réseau situés près de la paroi de développer une vitesse représentative de celle de la paroi, c'est-à-dire que l'interpolation de la vitesse des nœuds aux points de contrôle de la paroi devient similaire à la vitesse de la paroi à ces points, ce qui revient à établir une condition de glissement nul. Cette méthode permet de simuler le mouvement de particules dans un fluide ou d'étudier des problèmes d'interactions fluide-structure, dans la mesure où la distance entre les nœuds du réseau est beaucoup plus petite que la dimension de l'objet immergé.

1.1.5.2 Opérateur de collision

Un autre aspect important de l'évolution de la LBM implique l'opérateur de collision. Dans le modèle BGK tel que représenté à l'Éq. (1.29), cet opérateur ne contient qu'un seul coefficient de relaxation qui, pour un écoulement au nombre de Reynolds élevé, peut atteindre une valeur s'approchant de $\Delta t/\Delta t_R = 2$, ce qui correspond à une situation de sur-relaxation. Cette situation mène à des instabilités numériques et c'est pour pallier ce problème que d'autres opérateurs de collision ont été développés. Par exemple, d'Humières [30] a introduit un opérateur à relaxation multiple, communément désigné par l'acronyme MRT (Multiple Relaxation Time, en anglais), pour lequel on applique la relaxation aux moments des distributions plutôt que directement aux distributions. Ceci permet de séparer les échelles dynamiques du processus de collision et d'augmenter la stabilité numérique. Le processus de collision est donc représenté par $\mathbf{f}' = \mathbf{f} + \mathbf{M}^{-1} \mathbf{K} \mathbf{M} (\mathbf{f}^{eq} - \mathbf{f})$, où $\mathbf{f} = [f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9]$. Les distributions sont tout d'abord converties en moments au moyen de la matrice \mathbf{M} , puis ces moments sont relaxés par la

matrice des coefficients de relaxation \mathbf{K} . Finalement les moments sont reconverties en distributions par \mathbf{M}^{-1} .

Lätt a quant à lui proposé la méthode de régularisation [24], qui a pour objectif de forcer la méthode numérique à se rapprocher des dérivations de l'expansion Chapman-Enskog en remplaçant les parties non-équilibrées des distributions, f_i^{neq} , par leur approximation théorique $f_i^{(1)}$. Donc, avant de procéder à la séquence de collision-et-propagation, on régularise les distributions selon $\bar{f}_i = f_i^{eq} + f_i^{(1)}$, en ayant au préalable calculé les $f_i^{(1)}$ au moyen de l'expression

$$f_i^{(1)} = \frac{w_i}{2c_s^4} (c_{i\alpha} c_{i\beta} - c_s^2 \delta_{\alpha\beta}) \sum_k (f_k - f_k^{eq}) c_{k\alpha} c_{k\beta}, \text{ où la notation d'Einstein est utilisée. Bien que la}$$

collision effectuée sur les distributions \bar{f}_i suive le modèle BGK, la régularisation augmente la stabilité parce que les parties non-équilibrées des distributions de chacune des directions du réseau sont corrigées en tenant compte des distributions non-équilibrées de toutes les directions.

Une autre approche pour l'opérateur de collision, appelée entropique, a été proposée par Ansumali et Karlin [31] et est basée sur le théorème-H de Boltzmann, dont nous avons fait mention à la section 1.1.1. Cette méthode diffère de l'opérateur BGK car elle n'utilise pas une approximation polynomiale de la distribution de Maxwell-Boltzmann afin d'effectuer la collision, telle que représentée par l'Éq. (1.38), mais est plutôt construite à partir de la formulation d'une fonction $H(\mathbf{f})$ spécifique au réseau choisi. L'opérateur de collision a la forme suivante $\Delta^*(\mathbf{f}) = \beta \alpha(\mathbf{f}) \Delta(\mathbf{f})$, où la variation des distributions, $\Delta^*(\mathbf{f})$, dépend du coefficient β qui contrôle la viscosité et des fonctions $\alpha(\mathbf{f})$ et $\Delta(\mathbf{f})$ qui elles dépendent de la fonction H . Le terme de variation $\Delta(\mathbf{f})$ indique la direction de la collision dans l'espace des distributions et est le résultat de la résolution d'un système d'équations linéaires impliquant le gradient de la fonction H , $\nabla H(\mathbf{f})$. Un système d'équations doit ainsi être construit pour chaque nœud et chaque pas de temps du réseau. Quant à elle, la fonction $\alpha(\mathbf{f})$ détermine la projection maximale admissible de la variation $\Delta(\mathbf{f})$ afin que l'entropie ne diminue pas, ou que H n'augmente pas. Elle provient de la résolution de l'équation $H(\mathbf{f}) = H(\mathbf{f} + \alpha \Delta(\mathbf{f}))$, qui doit aussi être effectuée à chaque pas de temps. La stabilité de cet opérateur de collision provient de la limite imposée par α qui assure

qu'aucune distribution ne devienne négative après la collision. Par contre, cette méthode augmente substantiellement le temps de calcul en raison des équations à résoudre de façon continue.

Le dernier opérateur de collision présenté brièvement est celui de la collision en cascade de Geier [32,33]. Tout comme pour l'opérateur MRT, l'opérateur en cascade applique une relaxation à des moments des distributions basés sur les vitesses du réseau plutôt que directement sur les distributions. Or, il le fait d'une façon séquentielle plutôt que simultanément, d'où la désignation de cascade, en commençant par les moments d'ordres inférieurs. Pour ce faire, la relaxation de chaque moment est formulée de façon à ce qu'elle ne dépende que des relaxations des moments d'ordres inférieurs, ce qui est accompli en utilisant les états d'équilibre de moments centraux comme force motrice de la collision. Ces derniers ont comme origine la vitesse macroscopique locale et sont définis par $\kappa_{x^m y^n} = \sum_i f_i (c_{i,x} - u_x)^m (c_{i,y} - u_y)^n$. En d'autres termes, pour des moments

définis similairement à la méthode MRT, c'est-à-dire $\mathbf{m} = \mathbf{M}\mathbf{f}$, la méthode de collision en cascade applique successivement la relaxation à des variations de moments $\Delta\mathbf{m}$ dont les valeurs sont déterminées au préalable spécifiquement pour d'obtenir les moments centraux dans leur état d'équilibre selon $\kappa_{x^m y^n}^{eq} = \sum_i (f_i + \mathbf{M}^{-1}\Delta\mathbf{m})(c_{i,x} - u_x)^m (c_{i,y} - u_y)^n$. Le passage vers le domaine des

moments centraux permet le découplage des variations $\Delta\mathbf{m}$ d'ordres supérieurs avec leurs contreparties d'ordres inférieurs. Cette séparation des échelles dynamiques permet d'augmenter la stabilité numérique des simulations. Un autre ingrédient fondamental de cette méthode qui en accroît la stabilité est la définition des moments centraux d'équilibre. En effet, en intégrant dans l'espace continu des vitesses microscopiques les moments centraux de la distribution de Maxwell-Boltzmann, telle que représentée à l'Éq. (1.36) et ayant comme origine la vitesse macroscopique locale, les moments centraux obtenus sont constants et indépendants de la vitesse. La méthode de collision en cascade n'est donc pas basée sur une approximation polynomiale de la distribution de Maxwell-Boltzmann telle que montrée à l'Éq. (1.38) et valide seulement à proximité de la vitesse macroscopique nulle. Cette méthode est stable même lorsque les coefficients de relaxation appliqués aux variations de moments correspondants aux modes de transports hydrodynamiques s'approchent de 2, soit une situation de sur-relaxation sévère. De plus, même si elle requiert plus d'efforts de calculs que les méthodes BGK ou MRT, elle est explicite et plus simple que la méthode entropique, car elle ne nécessite pas la résolution d'équations.

1.1.5.3 Réseau à espacement variable

Le survol de l'évolution de la LBM sera complété en discutant des stratégies développées pour limiter le nombre de nœuds requis pour couvrir adéquatement un domaine de simulation. En effet, la simulation d'écoulements qui revêtent un intérêt pratique du point de vue de l'ingénierie nécessite généralement l'utilisation de maillages plus fins dans les zones où les gradients sont grands. Dans le contexte de la LBM utilisant un réseau régulier couvrant l'entièreté d'un domaine, cette nécessité de raffinement localisé amène à utiliser un réseau uniforme très fin sur tout le domaine, incluant les zones où c'est moins nécessaire. Ainsi, le nombre élevé de noeuds peut nécessiter l'utilisation de mémoire dépassant la capacité d'un ordinateur ou peut entraîner une exécution très lente des calculs. Afin de pallier ce problème, on peut avoir recours à une stratégie de domaines multiples, pour laquelle on raffine progressivement le réseau vers une zone spécifique. Chaque domaine ayant un espacement de réseau différent, et un pas de temps différent, après chaque cycle de collision-et-propagation un transfert d'information doit être exécuté avec les domaines voisins. Pour ce faire, les distributions f_i doivent être ajustées pour être cohérentes avec le coefficient de relaxation $\Delta t / \Delta t_R$ différent du réseau voisin. Lagrava et al. [34] ont proposé et validé la relation suivante: $f_i^A = f_i^{eq,A} + (\Delta t_R^A / \Delta t_R^B) f_i^{neq,B}$, où A et B désignent deux réseaux voisins. Cette méthode permet d'assurer une continuité des propriétés macroscopiques à la frontière des réseaux.

Par ailleurs, d'autres chercheurs ont travaillé à adapter la LBM à des réseaux orthogonaux non-uniformes. Ce type de réseau permet d'être adapté à la forme d'un objet, tel un cylindre ou un profil d'aile d'avion, et d'utiliser un raffinement variable dans l'espace. Par exemple, Imamura et al. [35] ont proposé la méthode de LBM par interpolation pour coordonnées généralisées, inspirée des travaux de He et Doolen [36]. Dans cette méthode, un réseau régulier est déformé dans le but d'être adapté à la géométrie du problème. Or, suite à cette transformation géométrique, lorsque les distributions sont propagées, leur point d'arrivée ne correspond plus avec un nœud du réseau. Afin de déterminer les distributions sur ces nœuds, il faut procéder à une interpolation spatiale. Cependant, afin de simplifier l'interpolation, qui nécessite d'être en amont au 2^e ordre, on l'exécute dans le domaine spatial du réseau régulier non déformé. Et afin de déterminer la correspondance entre les points d'interpolations des réseaux régulier et déformé, on utilise la matrice Jacobienne de transformations de coordonnées. Cette approche est intéressante mais est limitée à des

déformations géométriques simples. En contrepartie, des méthodes ont été proposées afin d'appliquer la LBM à des réseaux irréguliers, plus versatiles pour se conformer aux formes des objets et aux besoins de raffinement dans l'espace.

Plusieurs publications traitant de la combinaison de la méthode de volumes finis à la LBM sont disponibles dans la littérature. Selon cette approche, la formulation intégrale suivante de l'équation de Boltzmann à vitesses finies doit être respectée : $\int_{CV} (\partial f_i / \partial t + \mathbf{c}_i \cdot \nabla f_i - \Omega_i) dV = 0$, où CV représente le volume de contrôle choisi. Essentiellement cette équation stipule que la variation de la distribution f_i à l'intérieur du volume dépend des collisions se produisant dans tout le volume et, en appliquant le théorème de flux-divergence, des flux de distributions aux frontières du volume. Par exemple, dans la formulation de Stiebler et al. [37], chaque nœud du réseau est établi comme le centre d'un volume de contrôle et les frontières de ce volume sont positionnées à mi-chemin avec les nœuds voisins. Stiebler et al. assument que la variation temporelle d'une distribution est constante dans l'ensemble du volume et utilisent des interpolations linéaires aux frontières du volume pour calculer la contribution des collisions. Pour le terme convectif, ils utilisent une méthode de reconstruction de flux en amont afin d'augmenter la stabilité numérique. La stabilité semble d'ailleurs généralement poser un problème lorsqu'on veut augmenter le nombre de Reynolds des écoulements simulés par LBM en volumes finis.

Une alternative apparentée à la méthode de LBM en volumes finis a été développée récemment pour simuler des écoulements sur des maillages arbitraires, soit la méthode de résolveur de flux par LBM (lattice Boltzmann flux solver, en anglais). Cette méthode est en fait une méthode de volumes finis appliquée aux équations macroscopiques de continuité et de Navier-Stokes, mais qui introduit la LBM dans le calcul des flux aux frontières. Cette combinaison des deux approches est rendue possible par leur correspondance établie par la procédure de Chapman-Enskog mentionnée à la section 1.1.4, et plus spécifiquement par l'Éq. (1.44) qui fournit le tenseur de flux de quantité de mouvement (une quantité macroscopique) en fonction des distributions de l'espace mésoscopique. Ainsi, aux frontières des volumes de contrôles, on utilise les propriétés macroscopiques de densité et de vitesse pour passer au domaine mésoscopique des f_i^{eq} , puis après un processus local de propagation des distributions, on peut obtenir le tenseur de flux et repasser au domaine macroscopique pour exécuter l'intégration temporelle des volumes finis. Cette

méthode est plus stable numériquement que l'approche directe des volumes finis sur les distributions.

1.1.5.4 Pertinence pour la thèse

Les champs de recherche de la LBM que constituent les conditions aux frontières, les opérateurs de collision et l'utilisation de réseau à espacement variable ont été abordés spécifiquement parce que ce sont tous des ingrédients nécessaires à la construction de programmes servant à résoudre adéquatement des écoulements à hauts nombres de Reynolds, tels que ceux visés par la présente thèse. En effet, il est important d'utiliser des stratégies permettant d'appliquer précisément les conditions aux frontières sur des objets aux formes arbitraires, de s'assurer que le processus de collision soit stable même dans le cas d'une sur-relaxation des distributions s'approchant de la limite de 2 et de limiter le nombre total de nœuds du réseau tout en s'assurant d'avoir un espacement très fin près de l'objet.

1.2 Problématique

Cette section présente la problématique qui est à la base des travaux exécutés dans le cadre de cette thèse, dont les résultats seront présentés dans les sections subséquentes. Elle est le produit d'un intérêt pour la dynamique numérique des fluides appliquée à l'aérodynamique des ailes d'avions et d'une analyse de l'état de la recherche dans le contexte de l'application de la LBM aux écoulements turbulents sur profils d'ailes. Forcément, cette problématique est basée sur une revue critique de la littérature et sur une évaluation des lacunes existantes. La revue sera produite au chapitre suivant. Pour l'instant seulement que ses points saillants ne seront retenus.

1.2.1 Question de recherche

Au moyen de la présente thèse, on cherche simplement à répondre à la question principale suivante :

La méthode de Boltzmann sur réseau constitue-t-elle une approche adéquate pour simuler des écoulements turbulents sur profils d'ailes?

On cherche aussi une réponse à la question complémentaire suivante : Peut-on simuler les écoulements en utilisant des réseaux qui seraient aussi appropriés pour la considération éventuelle de profils altérés par la présence de glace, c'est-à-dire des réseaux qui peuvent fournir un raffinement adéquat pour des géométries arbitraires?

1.2.2 Lacunes actuelles

La revue de la littérature a révélé que des efforts de recherche avaient été déployés pour simuler le type d'écoulements qui nous intéresse au moyen de la LBM. Cependant, des lacunes ont été identifiées, auxquelles il est nécessaire de pallier afin de pouvoir établir la LBM comme alternative aux méthodes actuelles de simulation.

1. L'impact du choix d'opérateur de collision sur la stabilité numérique n'est pas abordé clairement;

L'opérateur de collision de type BGK étant reconnu comme instable pour les écoulements à hauts nombres de Reynolds, certains travaux produisent quand même des résultats pour lesquels la stabilité numérique n'a apparemment pas causé de problèmes mais n'indiquent pas clairement si une stratégie a été employée pour augmenter la stabilité. De plus, la stabilité de l'opérateur MRT pour des réseaux réguliers en configuration multi-domaine reste à être vérifiée. L'utilisation d'un opérateur plus stable, tel celui en cascade, n'avait pas été proposée pour la configuration multi-domaines lors du début de la présente thèse et ne l'a été qu'ultérieurement aux présents travaux.

2. L'utilisation du modèle de turbulence Spalart-Allmaras conjointement avec un réseau en configuration multi-domaine n'a pas été explorée adéquatement;

La configuration multi-domaine apparaît constituer un meilleur choix que certains autres types de maillages ne permettant pas de produire un raffinement localisé adéquat ou étant mal adaptés à la simulation d'un écoulement autour d'une géométrie hautement irrégulière, comme celle d'un profil d'aile givré. Le modèle de turbulence Spalart-Allmaras présente l'avantage de ne nécessiter la résolution que d'une seule équation, en comparaison de deux pour le modèle $k-\varepsilon$ par exemple. La seule étude répertoriée combinant ces deux modèles nous est apparue incomplète.

3. Le traitement numérique de la turbulence pour les profils bidimensionnels est peu détaillé;

Certains travaux ne font que mentionner quel modèle est employé sans décrire clairement une procédure permettant de reproduire l'implémentation du modèle. De plus, il n'y a pas d'étude qui traite de l'utilisation d'un modèle de turbulence avec une méthode LBM pour réseau non-structuré, lequel pourrait aussi présenter la versatilité nécessaire pour les profils à géométries irrégulières.

1.2.3 Hypothèse et buts

En regard des lacunes identifiées et de l'état de la recherche pour divers aspects de la LBM, l'hypothèse suivante est formulée: la LBM constitue une méthode adéquate de résolution d'écoulements turbulents sur profils d'ailes d'avion, dans la mesure où :

- on utilise un opérateur de collision adéquat pour assurer la stabilité à haut nombre de Reynolds, ou une alternative au processus de collision-et-propagation qui présente une stabilité numérique satisfaisante,
- on utilise une technique de raffinement de réseau qui permet d'obtenir une résolution adéquate près des parois d'un objet et qui serait appropriée pour un objet aux formes irrégulières,
- on résout une équation de transport de viscosité turbulente sur le même maillage ou réseau que celui employé pour la LBM.

Le but de la présente recherche consiste donc à concevoir et valider des programmes de simulations utilisant les dernières avancées en termes d'opérateur de collision stable ou de schéma autre que la collision-et-propagation, d'appliquer les techniques les plus récentes de résolution sur des réseaux multi-domaines ou non-structurés et d'y coupler adéquatement un modèle de turbulence. Un but connexe consiste à détailler clairement les travaux afin qu'un lecteur intéressé puisse reproduire les méthodes présentées.

1.2.4 Objectifs

Afin d'atteindre le but de la présente recherche une approche séquentielle est adoptée, en établissant des objectifs distincts pour chaque partie du travail. De plus, comme le format retenu pour le travail est celui de la thèse par article, un article a été rédigé pour chaque objectif principal

de la recherche. Ces articles sont fournis en annexe et une synthèse de l'ensemble du travail est produite au chapitre 3.

Objectif 1 Développer les connaissances de la LBM et les aptitudes à créer des programmes de résolution d'écoulements fluides utilisant cette méthode. Développer une idée nouvelle servant de base au premier article.

Sous-objectif 1 Créer un programme LBM en 2D pour écoulements incompressibles dans des domaines ouvert ou fermé, avec ou sans objet à l'intérieur.

Sous-objectif 2 Développer un nouveau modèle de conditions aux frontières et l'inclure dans le programme 2D.

Sous-objectif 3 Valider le programme avec des cas typiques répertoriés dans la littérature et comparer le nouveau modèle de condition aux frontières avec des modèles existants.

Objectif 2 Développer un programme LBM basé sur une approche de réseau multi-domaines et l'appliquer à la résolution d'écoulements sur profils d'ailes.

Sous-objectif 1 Créer le programme en y incluant des techniques récentes de transfert d'information entre les différents niveaux de réseaux. Écrire le code de telle sorte qu'un nombre arbitraire de réseaux distincts puissent être utilisés. Inclure au programme un opérateur de collision stable et le modèle de conditions aux frontières développé pour le premier objectif du présent travail.

Sous-objectif 2 Choisir un modèle de turbulence approprié dans la littérature et formuler une méthode de résolution adaptée au réseau multi-domaine. Programmer cette méthode.

Sous-objectif 3 Choisir dans la littérature des résultats numériques obtenus avec des résolveurs Navier-Stokes et des résultats obtenus expérimentalement. Simuler les mêmes écoulements avec notre programme et comparer les résultats.

Objectif 3 Développer un programme LBM basé sur un réseau non-structuré et l'appliquer à la résolution d'écoulements sur profils d'ailes.

Sous-objectif 1 Déterminer le modèle LBM pour réseau non-structuré apparaissant comme étant le plus adéquat pour le contexte du présent travail et créer un programme basé sur ce dernier.

Sous-objectif 2 Formuler une méthode de résolution adaptée au réseau non-structuré pour le modèle de turbulence sélectionné pour le deuxième objectif du présent travail. Programmer cette méthode.

Sous-objectif 3 Simuler avec le programme des écoulements repris des travaux exécutés pour le deuxième objectif et comparer les résultats.

1.2.5 Nouveautés

Le présent travail n'a pas comme objectif d'approfondir les fondements théoriques de la LBM. Il est plutôt abordé avec le pragmatisme d'un ingénieur possédant une expérience industrielle, intéressé à répertorier et combiner les dernières avancées dans le domaine de la LBM afin de les appliquer à un problème pratique. Néanmoins, cette thèse comporte des nouveautés pouvant certainement stimuler l'intérêt de la communauté LBM et contribuer au développement de la LBM dans le contexte de l'aérodynamique.

Le modèle de conditions aux frontières développé pour le premier objectif constitue la première nouveauté du travail. Tel que nous le verrons au chapitre 3.1, ce modèle déroge de certaines assumptions utilisées par d'autres modèles et présente des avantages pour les écoulements à hauts nombres de Reynolds.

Un aspect du travail pour lequel aucun précédent satisfaisant n'a été trouvé dans la littérature est la combinaison d'un résolveur LBM pour réseau de type multi-domaines avec le modèle de turbulence Spalart-Allmaras, qui a été accomplie dans le cadre du deuxième objectif du travail. Ce modèle a certainement déjà été utilisé par un résolveur de type Navier-Stokes sur un maillage à domaines multiples, mais dans le contexte LBM nous n'avons observé son utilisation adéquate que pour un réseau de type orthogonal à espacement variable.

Finalement, l'utilisation de la méthode de résolveur de flux par LBM, développée récemment, pour simuler des écoulements turbulents sur profils d'ailes au moyen de réseaux non-structurés constitue une nouveauté intéressante. Bien que cette approche déroge d'une application «pure» de

la LBM en formant un hybride avec la méthode des volumes finis, elle permet d'employer des réseaux non-structurés qui présentent certainement l'avantage d'avoir une grande adaptabilité.

CHAPITRE 2 REVUE CRITIQUE DE LA LITTÉRATURE

Le thème principal de cette thèse est la LBM appliquée aux écoulements incompressibles turbulents sur profils d'ailes. Ce sera donc le premier sujet abordé dans cette revue. Or, comme dans la littérature des travaux ont aussi été publiés sur les écoulements turbulents pour des ailes tridimensionnelles, cet aspect sera aussi inclus dans la première section de la revue.

Ensuite, comme le premier objectif du travail inclut le développement d'un nouveau modèle de conditions aux frontières, cet aspect de la LBM sera abordé dans la deuxième section de la revue.

Certains autres aspects connexes de la LBM sont aussi pertinents à la thèse, comme les opérateurs de collision et les méthodes de raffinement de réseau. Or, puisque ces thèmes ont été couverts en introduction et que le présent travail ne vise pas à développer de nouvelles méthodes, ils ne feront pas partie de la revue de littérature.

2.1 Écoulements turbulents sur ailes

Afin de simuler numériquement des écoulements turbulents sur des ailes (3D) ou des profils d'ailes (2D), il existe essentiellement trois options pour traiter de la turbulence. La première option, applicable en 3D, consiste à utiliser un maillage suffisamment raffiné pour couvrir toutes les échelles dynamiques de la turbulence. Dans la littérature, cette méthode est appelée «simulation numérique directe» (Direct Numerical Simulation ou DNS, en anglais). Son inconvénient majeur est évidemment le nombre très élevé de nœuds requis, dont l'espacement minimum est inversement proportionnel au nombre de Reynolds. Dans le contexte de la LBM, la méthode DNS n'a été utilisée jusqu'ici que pour étudier la propagation d'ondes acoustiques autour d'une aile au profil à diffusion contrôlée [38,39]. L'étude a été effectuée au moyen du logiciel commercial PowerFlow en utilisant une approche de multi-domaines et en limitant la profondeur du domaine numérique en employant des conditions aux frontières de périodicité. Le réseau était composé de 640M de volumes cubiques reliant les nœuds et la simulation d'une seconde de temps réel a requis plus de 10 jours de calcul à 512 processeurs. Les résultats obtenus, tant au point de vue des coefficients de pression, des profils de vitesse que de l'amplitude des ondes sonores concordaient bien avec des résultats expérimentaux. Cette étude montre que, dans la mesure où on fait abstraction de la nécessité d'utiliser un réseau au nombre très élevé de nœuds, voire irréaliste pour les capacités computationnelles actuelles, la LBM peut résoudre des écoulements sans avoir recours à des

modèles de turbulence. Il faut par contre mentionner que les auteurs indiquent que le logiciel PowerFlow maintient la stabilité numérique en augmentant localement la viscosité utilisée dans le calcul du coefficient de relaxation du modèle de collision BGK, sans toutefois en spécifier les détails.

La deuxième option de traitement de la turbulence en 3D, qui représente une alternative à la méthode DNS, est appelée simulation des grandes échelles (Large Eddy Simulation ou LES, en anglais). Comme son nom l'indique, cette méthode ne vise pas à simuler toutes les échelles de turbulence mais seulement celles qui sont plus grandes que l'espacement du réseau. L'écoulement résolu est donc instationnaire car le «filtre» n'atténue pas les tourbillons de plus grande amplitude. Cette méthode emploie un modèle de viscosité turbulente dont l'impact est de réduire le coefficient de relaxation employé dans la LBM, comme par exemple le modèle bien connu de Smagorinsky. L'approche de LES est utilisée par les logiciels commerciaux XFlow [40] et PowerFlow [41], avec des modèles de viscosité turbulente qui ne seront pas expliqués en détail ici. Cette approche, qui limite le nombre total de nœuds comparativement à l'approche DNS, a permis d'utiliser ces logiciels commerciaux sur des super-ordinateurs pour simuler l'écoulement autour du modèle aile/fuselage DLR F11. Même si les résultats obtenus en termes de coefficients de portance et de traînée ne reflètent pas parfaitement les valeurs expérimentales [40,41], les simulations permettent tout de même de recueillir de l'information pertinente sur les zones de transition de écoulement laminaire vers turbulent et sur la propagation des tourbillons sur la surface de l'aile. Or, avec des temps d'exécution de l'ordre d'un jour ou deux, on constate que ce type de simulation nécessite beaucoup de ressources computationnelles.

Le logiciel PowerFlow et son modèle LES a aussi été utilisé pour étudier la formation de tourbillons en bout d'ailes [42], l'écoulement instationnaire sur une aile à multi-surface [43], la séparation de l'écoulement au bord de fuite d'une aile de type NACA 4412 [44] et plus récemment l'écoulement sur un profil NACA23012 altéré par la formation de glace [45]. Pour cette étude, la simulation 3D permet d'observer le caractère tridimensionnel de la séparation de l'écoulement induite par la présence de glace de type «glaze ice» qui forme des cornes sur le bord d'attaque de l'aile. Les auteurs rapportent avoir obtenu une bonne concordance des mesures aérodynamiques avec des résultats expérimentaux. L'approche LBM-LES utilisée par le logiciel PowerFlow a aussi été étendue aux écoulements transsoniques et une simulation a notamment été exécutée pour une aile de type Onera M6 [46].

La troisième option pour le traitement de la turbulence, applicable en 2D et en 3D, consiste à en modéliser toutes les échelles. Cette méthode a aussi recours à une viscosité turbulente, or son calcul n'est pas seulement local, comme pour plusieurs méthodes LES, mais elle est plutôt le produit de la résolution d'une équation de transport. Cette viscosité est donc générée, transportée, diffusée et détruite selon les conditions locales de l'écoulement. Il en résulte un état stationnaire de l'écoulement où tous les tourbillons associés à la turbulence sont supprimés pour laisser place à un écoulement moyen. Comme la turbulence est un phénomène tridimensionnel, son traitement en 2D nécessite d'en traiter toutes les échelles afin d'atténuer les tourbillons qui n'auraient pas la même signification physique qu'en 3D, car la restriction d'une dimension limiterait la quantité de mouvement qu'ils transporteraient. Dans les approches de dynamique numérique des fluides (Computational Fluid Dynamics ou CFD, en anglais), ce type de modélisation se fait au moyen de équations de Navier-Stokes moyennées (Reynolds-Averaged Navier-Stokes equations ou RANS, en anglais) où un terme de contrainte visqueuse apparaît et sert à insérer la viscosité turbulente.

Or, dans le contexte de la LBM, on peut se demander pourquoi considérer la troisième option si, comme il a été mentionné, un logiciel commercial existe et permet déjà d'étudier les écoulements incompressibles turbulents en 3D, y compris sur une aile givrée. Il y a deux éléments de réponse à cette question. Tout d'abord, il y a la limitation des ressources. Les simulations 3D nécessitent encore l'utilisation de super-ordinateurs ou du moins d'une architecture parallèle de processeurs, qui n'est pas toujours disponible. Deuxièmement, dans une phase préliminaire de conception de profil d'aile ou d'analyse aérodynamique de profil altéré par de la glace, on peut être intéressé à déterminer les caractéristiques aérodynamiques sans nécessairement désirer de détails sur la structure tridimensionnelle d'un écoulement. Cela constitue une première approximation obtenue avec moins de ressources, car un ordinateur peut suffire. C'est d'ailleurs dans ce contexte que s'inscrit cette thèse.

Les premières études répertoriées d'écoulements turbulents sur profil d'aile en LBM ont été effectuées par Filippova et al. [47] et Succi et al. [48]. Elles visaient à appliquer la première méthode développée de raffinement de réseau par méthode de multi-domaines au contexte d'un écoulement aérodynamique, tout en y joignant la résolution par différence finie des équations du modèle de turbulence $k-\varepsilon$. Pour des nombres de Reynolds de l'ordre de 10^6 , ils ont obtenu des coefficients de pression pour un profil NACA 4412 concordant généralement bien avec des données expérimentales. Ces deux études ont utilisé l'opérateur de collision BGK, sans toutefois

rapporter d'instabilité numérique ni si une méthode particulière a été employée pour limiter ces dernières.

Ensuite, Imamura et al. [35] ont simulé l'écoulement autour d'un profil NACA 0012 au moyen de la LBM par interpolation. Ils ont employé un réseau structuré déformé pour s'adapter au contour du profil, communément appelé C-grid. Ils ont aussi fait usage du modèle de turbulence Baldwin-Lomax. Pour un écoulement à $Re=5 \times 10^5$ et un angle d'attaque de 7° , ils ont rapporté avoir obtenu des coefficients de portance et de traînée presque identiques à ceux provenant du logiciel établi de la NASA CFL3D. Or, la différence des résultats apparaît plutôt être non-négligeable, particulièrement pour la portance, alors que le coefficient est inférieur de 6%. De plus, les auteurs ne fournissent pas de détails sur l'implémentation particulière du modèle Baldwin-Lomax sur leur type de réseau. Ils réfèrent plutôt à Teixeira [49] pour la procédure, mais cette source indique seulement comment utiliser la viscosité turbulente pour modifier le coefficient de relaxation. Finalement, aucune instabilité numérique n'est rapportée bien que le coefficient de relaxation pour la collision approche forcément la limite de 2, alors on peut se demander si la méthode par interpolation est intrinsèquement dissipative. Premnath et al. [50] affirment justement que les méthodes par interpolation peuvent introduire une dissipation numérique considérable.

Li et al. [51] ont utilisé le logiciel 3D PowerFlow dans une configuration quasi-2D pour simuler l'écoulement autour d'un profil d'aile d'avion d'affaire GLC 305 modifié par la présence de glace de type «rime ice». L'approche multi-domaine a été utilisée, incluant le modèle de transfert d'information entre les niveaux de réseaux voisins développé pour ce logiciel [52,53]. De plus, ils ont modélisé la turbulence par une méthode de différence finie appliquée aux équations d'une version modifiée du modèle $k-\epsilon$. Pour le profil sans givre et un nombre de Reynolds de 7.5×10^6 , les coefficients de portance et de pression obtenus concordent bien avec des résultats expérimentaux. Pour le profil givré à $Re=3.5 \times 10^6$, le décrochage aérodynamique est obtenu à un angle d'attaque inférieur à celui des données expérimentales et le coefficient de traînée est généralement moins précis. Il faut aussi mentionner que les auteurs ont fait usage de fonctions de paroi afin de limiter le nombre de nœuds dans la couche limite. L'inconvénient de cette étude est que le modèle de turbulence choisi nécessite la résolution de deux équations différentielles et que l'implémentation numérique n'est pas détaillée. Le logiciel PowerFlow a aussi été utilisé par Chi et al. [54] pour comparer les écoulements autour du profil GLC 305 givré avec ceux obtenus par d'autres logiciels. Les détails des résultats ne seront pas discutés, mais il faut mentionner que dans cette étude le

maillage utilisé par les autres logiciels, de type C-grid orthogonal courbé pour se conformer au profil givré, apparaît beaucoup plus complexe à générer qu'un réseau régulier pour la méthode LBM. En ce sens l'approche de réseau multi-domaine semble avantageuse, d'autant plus qu'on peut appliquer les conditions aux frontières du profil d'aile sans que les nœuds du réseau soient situés directement sur la paroi.

L'approche de raffinement de réseau par multi-domaine selon le modèle développé pour PowerFlow [53] a été reprise par Chen [55] pour simuler l'écoulement autour d'un profil NACA 0012 à $Re=10^5$ et un angle d'attaque de 4° . Il a aussi employé le modèle de collision MRT et le modèle de turbulence à une équation de Spalart-Allmaras. À priori cela semble constituer une approche intéressante, car l'opérateur MRT augmente la stabilité numérique et le modèle S-A contient moins d'équations que le modèle $k-\epsilon$. Or, ce travail présente des lacunes. Premièrement, l'auteur ne fournit aucun détail de son implémentation du modèle S-A. Deuxièmement, les résultats obtenus sont discutables. L'auteur indique bien capturer le détachement de l'écoulement et apparaît obtenir une solution instationnaire. Pour les conditions de l'écoulement et l'utilisation du modèle S-A, ce détachement est fortement improbable. Comme l'espacement du réseau près du profil n'est que de $1/800^\circ$ de corde, le réseau utilisé n'est probablement pas assez raffiné et en ce sens l'étude est incomplète.

L'utilisation combinée du modèle de collision MRT et du modèle de turbulence Spalart-Allmaras a aussi été utilisée par Li et al. [56]. Cependant, leur approche de raffinement de réseau est plutôt un hybride entre celles de Chen [55] et de Imamura et al. [35]. En effet, leur réseau ne se conforme pas au profil mais le couvre de façon à ce qu'il se trouve immergé, comme c'est le cas pour Chen [55]. De plus, ils n'ont pas utilisé l'approche multi-domaine mais plutôt celle d'un réseau cartésien déformé afin de produire un meilleur raffinement près du profil, comme pour Imamura et al. [35]. Par contre, leur réseau, qu'ils ont désigné « non-body-fitted », conserve un alignement purement rectiligne des nœuds avec des rangées et colonnes plus rapprochées à proximité du profil que lorsqu'on s'en éloigne. Ils ont aussi adopté l'approche de la LBM par interpolation, appropriée pour ce type de réseau. Pour un profil NACA 0012 à des angles d'attaque de 0° et 7° , ils ont obtenu des profils de coefficients de pression et de vitesse démontrant une bonne concordance avec des résultats numériques du logiciel CFL3D. La qualité de leurs résultats surpasse celle de Chen [55] entre autres parce que l'espacement minimum du réseau à proximité du profil est de $1/5000^\circ$ de corde. Or, bien que ce niveau de raffinement apparaisse comme étant adéquat, il est obtenu aux

dépens d'un nombre de nœuds très élevé, soit plus de 4 millions. Pour cette raison, nous considérons que ce type de réseau est mal adapté, d'autant plus que si la géométrie du profil était plus arbitraire, ou que le profil était plus incliné que 7° , il serait difficile de raffiner plus localement sans que le nombre total de nœuds n'augmente de façon très significative. Par ailleurs, Li et al. [56] ne fournissent pas non plus de détails sur l'implémentation du modèle S-A. Ils réfèrent plutôt à un article qui implémente ce modèle en 3D sur maillage multi-domaine pour un écoulement compressible, à l'article original du modèle S-A et au manuel de CFL3D. Finalement, bien que le modèle de collision MRT soit plus stable que le BGK, on peut se demander encore une fois si la méthode LBM par interpolation a un caractère dissipatif qui contribue à la stabilité. À priori, les travaux de Chen [55] indiquent que le modèle MRT serait stable sur un réseau régulier mais étant donné la qualité de ses résultats, quelques réserves peuvent être émises.

Une autre étude d'écoulement sur profil bidimensionnel, effectuée par Si et Shi [57], laisse perplexe en raison du choix de modèle de turbulence. Les auteurs ont choisi de simuler l'écoulement autour d'un profil NACA 0012 en combinant un réseau bidimensionnel régulier, l'opérateur de collision BGK et le modèle de turbulence LES Smagorinsky. Comme il a été mentionné précédemment qu'en 2D il est nécessaire de modéliser toutes les échelles de turbulence, le choix du modèle LES apparaît donc inapproprié. Pour un écoulement à $Re=5 \times 10^5$ et un angle d'attaque de 7° , les auteurs obtiennent un coefficient de portance inférieur de seulement 4% à celui de CFL3D, or ils ne mentionnent pas si ce coefficient a été calculé par une moyenne temporelle. Pour leurs choix de modèles, on pourrait s'attendre à ce que l'écoulement sur la partie supérieure du profil soit instationnaire mais l'article ne le précise pas. De plus, aucune donnée n'est fournie sur l'espacement du réseau.

La dernière et plus récente étude qui sera mentionnée, soit celle de Ishida [58], présente une ambiguïté similaire à celle de Si et Shi [57]. C'est-à-dire qu'on y simule un écoulement bidimensionnel turbulent sur un profil d'aile à éléments multiples sans avoir recours à un modèle de turbulence. La solution obtenue est donc instationnaire et les coefficients de pression rapportés sont des mesures moyennées temporellement. La méthode proposée s'apparente donc à une méthode de LES, à la différence qu'aucune viscosité turbulente n'est ajoutée. La stabilité numérique est d'ailleurs assurée par l'utilisation de l'opérateur de collision en cascade. Cette étude présente par ailleurs l'aspect intéressant d'unir cet opérateur de collision stable à une configuration multi-domaines pour le réseau, même si le détail des algorithmes n'est pas fourni. Cette dernière

observation est particulièrement pertinente dans le contexte de la présente thèse et il faut souligner que le travail de Ishida [58] a été publié en 2016.

La recension des études indique qu'il y a des lacunes dans la recherche que nous pouvons essayer de combler. Tout d'abord, lors du début des présents travaux, la stabilité numérique des opérateurs de collision dans le contexte des écoulements turbulents autour de profils d'ailes demeurerait une question ouverte. Les études utilisant l'opérateur BGK [47,48,59] n'indiquent pas qu'une mesure particulière ait été prise pour l'augmenter ou que l'ajout de viscosité turbulente ait contribué à cette stabilité. Pour ce qui est de l'opérateur MRT, sa stabilité demeure à confirmer lorsqu'utilisé sur un réseau régulier, car les résultats de Chen [55] ne sont pas convaincants. De plus, lorsqu'un réseau adapté à la LBM par interpolation est employé [56], la stabilité de l'opérateur MRT est peut-être augmentée artificiellement par les interpolations qui dissipent certaines échelles dynamiques. Comme ce caractère dissipatif n'est pas présent pour les réseaux réguliers, il n'apparaît pas clair que l'opérateur MRT soit stable dans ce contexte. Par ailleurs, l'utilisation d'un opérateur de collision plus stable n'a été proposée que bien ultérieurement au début de cette thèse. En deuxième lieu, l'utilisation conjointe du modèle de turbulence Spalart-Allmaras et d'un réseau en configuration multi-domaine n'a pas été explorée adéquatement. Puisque le modèle S-A présente l'avantage de n'être constitué que d'une seule équation, et que le réseau multi-domaine est plus approprié que les autres types de maillages recensés pour s'adapter aux profils d'ailes à géométrie irrégulière, la combinaison des deux nous semble être un bon choix. Troisièmement, les procédures numériques de l'implémentation des modèles de turbulence sont peu détaillées et ne permettent pas nécessairement de reproduire les travaux des auteurs. De plus, aucune étude ne traite de l'utilisation d'un modèle de turbulence avec une méthode LBM pour réseau non-structuré. Ce type de réseau est très versatile et peut aussi bien s'adapter aux profils à géométries irrégulières.

2.2 Conditions aux frontières sur surface solide courbe

Puisque le premier objectif de la thèse est de développer les connaissances de la LBM, incluant le développement d'un nouveau modèle de conditions aux frontières, une revue des méthodes existantes dans ce domaine est effectuée. Il existe une panoplie de méthodes pour appliquer les conditions aux frontières sur surface solide courbe. Le sujet a d'ailleurs été abordé brièvement en introduction (section 1.1.5.1).

2.2.1 Approche générale par réflexion

Comme la LBM utilise un réseau structuré, le traitement des frontières courbes est basé sur la distance Δ entre la frontière et les noeuds l'avoisinant. Il faut déterminer les distributions qui sont en quelque sorte reflétées par la frontière. La première méthode développée pour les frontières courbes fut proposée par Filippova et Hänel [60] pour simuler un écoulement autour de fibres sur lesquelles s'accumulent des particules. Selon leur méthode, la distribution reflétée par la frontière est calculée au moyen d'une combinaison linéaire de la distribution propagée vers la paroi par le noeud voisin, de la distribution à l'équilibre sur ce noeud et de la distribution à l'équilibre sur le noeud solide (noeud du réseau situé de l'autre côté de la frontière) en direction inverse. La distribution à l'équilibre sur le noeud solide est établie au moyen d'une extrapolation de la vitesse à partir des vitesses du noeud voisin et de la frontière. La méthode est représentée par la relation suivante $f_{i'} = (1 - w_i) f_i + a_1 w_i f_i^{eq} + a_2 f_{i',solide}^{eq}$, où les coefficients w_i , a_1 et a_2 dépendent de la distance Δ entre le noeud et la frontière.

Or, Mei et al. [61] ont noté que la méthode de Filippova et Hänel [60] induisait une forte instabilité lorsque Δ était inférieur à 0.5. Pour remédier à cette situation, ils ont amélioré la méthode essentiellement en utilisant la vitesse du deuxième noeud voisin de la frontière pour l'extrapolation de la vitesse sur le noeud solide lorsque $\Delta < 0.5$. Ils ont vérifié leur méthode au moyen de cas tests, tel un écoulement dans un canal induit par différence de pression, et ont conclu que l'erreur du champ de vitesse réduisait en fonction de l'espacement du réseau avec un ordre du 2^e degré. Par la suite, Guo et al. [28] ont proposé une nouvelle méthode qui était, selon leur conclusion, plus stable numériquement que les deux méthodes mentionnées précédemment. Tout comme Mei et al. [61], ils ont utilisé une extrapolation de la vitesse sur le noeud solide au moyen des vitesses de la frontière et des noeuds voisins. Cependant, leur traitement différait par l'assomption que la partie non-équilibrée des distributions sur le noeud solide, avant le processus de collision-et-propagation, pouvait être approximée par celles des noeuds voisins de la frontière. Ainsi, disposant à la fois des distributions à l'équilibre et non-équilibrée sur les noeuds solides, on peut exécuter la collision sur ces noeuds et ensuite propager les distributions sur les noeuds fluides: $f_{i'} = f_{i',solide}^{eq} + (1 - \Delta t / \Delta t_R) f_{i',solide}^{neq} = f_{i',solide}^{eq} + (1 - \Delta t / \Delta t_R) f_{i',fluide}^{neq}$. Ils ont démontré que leur méthode produisait des résultats avec une exactitude du 2^e ordre en simulant un écoulement de Couette entre cylindres concentriques.

Pendant la même période, Bouzidi et al. [27] ont proposé une méthode alternative ne nécessitant pas l'extrapolation de la vitesse sur le noeud solide. Ils proposèrent plutôt de déterminer les distributions manquantes par interpolation, c'est-à-dire que pour une distance Δ et une période Δt données, ils interpolent les distributions connues à l'endroit duquel émanerait la distribution qui arriverait exactement sur le noeud voisin de la frontière après réflexion sur cette frontière. Pour une vitesse non-nulle de la frontière, on ajoute ou soustrait à cette distribution une contribution δf qui dépend de la distance Δ . Ils ont aussi démontré une exactitude du 2^e ordre au moyen de l'écoulement de Couette entre cylindres concentriques. Leur méthode a ensuite été adaptée par Lallemand et Luo [62] afin d'étudier les écoulements sur des frontières mobiles. Ils ont étudié le mouvement d'un cylindre dans différents types d'écoulements et ont conclu que, bien que les forces exercées sur le cylindre en mouvement puissent être calculées avec une bonne exactitude, la conservation de la masse près de la frontière en mouvement demeure un problème lorsqu'un noeud passe de l'intérieur vers l'extérieur du cylindre, et vice-versa. Rohde et al. [63] ont aussi observé que bien que la méthode de Bouzidi et al. [27] fournisse une exactitude du 2^e ordre pour le champ de vitesse, elle ne conserve pas la masse.

Ginzburg et d'Humières [64] ont proposé la méthode de multi-réflexion afin de faire passer l'exactitude de la méthode proposée par Bouzidi et al. [27] au 3^e ordre. Pour ce faire, leur méthode utilise l'information du 3^e noeud voisin. La conclusion principale émanant de la validation numérique qu'ils ont effectuée pour une série de cas test, incluant le déplacement d'un cylindre entre deux parois, est que leur méthode produit des champs de vitesse et de pression plus précis à proximité de la paroi que les autres types de conditions aux frontières proposés précédemment. Or, Chun et Ladd [65] ont observé que pour certains types d'écoulements la méthode de multi-réflexion est difficilement applicable car trois noeuds fluides ne sont pas toujours disponibles. Ils ont développé une méthode basée sur l'interpolation des distributions à l'équilibre utilisant moins de noeuds fluides pour l'interpolation. Pour y arriver, ils ajoutent un point d'interpolation sur la paroi en utilisant la distribution à l'équilibre pour la vitesse prescrite sur la paroi. Ils suggèrent que leur méthode est applicable particulièrement aux écoulements dans des géométries étroites, mais notent qu'elle ne conserve pas parfaitement la masse. Cependant, ils affirment aussi que bien que d'autres auteurs aient développé des méthodes de conditions aux frontières par interpolation qui conservent la masse [66,67], elles sont plus difficile à implémenter et n'ont généralement pas une exactitude

du 2^e ordre. Il faut par contre noter que les auteurs de ces dernières études soutiennent plutôt le contraire et que l'exactitude du 2^e ordre est obtenue.

Plus récemment, d'autres approches ont été proposées pour traiter des conditions sur frontières courbes. En rappelant une observation faite par Lallemand et Luo [62] selon laquelle les méthodes de conditions aux frontières utilisant l'interpolation ne conservent pas la masse près des frontières, Kao et Yang [68] ont proposé une méthode sans interpolation basée sur les travaux de Filippova et Hänel [69] pour les réseaux multiples. Ils ont appliqué le principe de continuité de contrainte (stress) permettant de transférer l'information d'un réseau à un réseau sous-jacent plus raffiné, en assumant que la distance séparant un noeud d'une frontière courbe était analogue à traiter la situation comme un transfert d'information entre réseaux. En étudiant l'écoulement autour d'un cylindre carré, ils ont conclu que leur méthode conservait mieux la masse que les méthodes par interpolation qu'ils ont testées.

Kuo et Chen [70] ont aussi proposé une méthode n'utilisant pas d'interpolation, mais plutôt basée sur ce qu'ils appellent le « tangential momentum accomodation coefficient » qui permet de contrôler la vitesse de glissement tangentielle sur une paroi. Or, bien que le concept soit intéressant, ils ne l'ont appliqué qu'à des frontières droites, donc il perd de la pertinence dans le cadre du présent projet.

Kang et al. [71] ont proposé une amélioration au modèle de Guo et al. [28] essentiellement en ajoutant un terme de correction de vitesse servant à calculer les distributions à l'équilibre sur le noeud solide. Cette correction provient de la différence entre la vitesse désirée sur la paroi et la vitesse y étant mesurée, par extrapolation, et utilisée comme « feed-back ». Or, dépendamment de la position de la paroi, l'extrapolation de la vitesse peut nécessiter l'utilisation de trois noeuds fluides. Cette méthode possède donc la même limitation que la méthode de Ginzburg et d'Humières [64] en ce sens que toutes les géométries ne permettent pas l'utilisation d'autant de noeuds.

Tiwari et Vanka [72] ont aussi proposé une méthode qui est en fait une variante de la méthode de Guo et al. [28], c'est-à-dire que les distributions manquantes post-propagation proviennent des noeuds solides situées de l'autre côté de la frontière. Or, Tiwari et Vanka n'extrapolent pas la vitesse à partir des noeuds fluides voisins dans une direction spécifique du réseau, mais ils utilisent plutôt une direction normale à la frontière et y positionnent des points où la vitesse est interpolée à partir des noeuds fluides environnants. Ces vitesses interpolées servent ensuite dans une

extrapolation permettant d'établir les distributions à l'équilibre sur les nœuds solides. De plus, ils extrapolent aussi les distributions non-équilibrées à partir de la direction normale. Tiwari et Vanka [72] rapportent une exactitude du 2^e ordre pour un écoulement de Couette entre cylindres concentriques. Des modifications à cette méthode ont aussi été apportées par Chen et al. [73] afin de réduire les ondes parasites de pression induites près de la frontière lorsque celle-ci est en mouvement sur le réseau.

Chen et al. [74] ont quant à eux utilisé une autre approche, basée sur la chimie des surfaces, pour développer une méthode spécifiquement pour conserver la masse. Ils utilisent la surface comme accumulateur de masse, qui emmagasine et libère de la masse selon les conditions locales. De plus, l'information d'un seul nœud fluide n'est utilisée pour calculer une distribution manquante post-propagation. Ils ont démontré que cette méthode produit une exactitude numérique du 2^e ordre et conserve la masse totale pour le cas d'un écoulement de Couette entre cylindres concentriques. Cette méthode est très intéressante car elle utilise un minimum d'information dans le fluide (un seul nœud), qu'elle conserve la masse et qu'elle ne nécessite pas d'assomption quant aux parties non-équilibrées des distributions. Or, elle n'a pas été appliquée aux cas de frontières en mouvement, c'est-à-dire si un objet se déplace sur le réseau, et il apparaît problématique de déterminer comment les nœuds frontières servant d'accumulateurs de masse seraient traités dans un tel cas. Cette constatation quant aux frontières en mouvement nous amènera plus loin sur le sujet des conditions aux frontières immergées dans le contexte de la LBM.

2.2.2 Approche volumétrique

Pour plusieurs méthodes de conditions aux frontières mentionnées précédemment, basées sur la propagation de distributions de particules issues de processus d'interpolations, un inconvénient souvent identifié est que la conservation de masse n'est pas respectée parfaitement. Afin d'assurer le respect de ce principe de conservation, Chen et al. [75] ont développé la méthode de conditions aux frontières volumétriques. Le principe général de la méthode est de réfléchir et redistribuer sur les nœuds voisins d'une section de frontière solide toutes les particules recueillies par cette section pendant un pas de temps. En ce sens, cette méthode constitue en fait une classe de méthode par réflexion. L'adjectif volumétrique provient du fait que la quantité de particules recueillies par une surface est calculée au moyen des distributions dans les cellules interceptées par un volume couvrant toutes les particules pouvant atteindre la surface pendant un pas de temps. Par exemple,

pour une surface verticale d'aire A , bordée à gauche par le fluide et à droite par le solide, toutes les distributions f_2 pouvant atteindre la surface pendant un pas de temps Δt se trouvent dans un volume $|\mathbf{c}_2 \cdot \mathbf{n}_A| A \Delta t$ situé à la gauche de la surface, où \mathbf{n}_A est le vecteur normal à la surface. Les distributions sont ensuite réfléchies et redistribuées proportionnellement à la fraction de volume occupée par les cellules interceptées. Cette méthode permet aussi d'imposer des conditions de glissement mais les détails de la procédure ne seront pas abordés ici.

Cette méthode a été reprise par Rohde et al. [67] pour simuler l'écoulement autour d'objets en mouvement. Or des lacunes à cette méthode ont été identifiées. Li et al. [52] ont indiqué que la méthode souffrait de diffusion numérique près des frontières, particulièrement lorsque cette dernière était orientée arbitrairement relativement au réseau. Rohde et al. [63] et Rohde [76] ont aussi observé que la méthode ne démontrait une exactitude que du 1^{er} ordre. Pour ces raisons, des versions améliorées de la méthode ont été proposées par Li [77], Li et al. [78] et Rohde et al. [63]. Un autre inconvénient de cette méthode est la complexité du traitement géométrique nécessaire au calcul des interceptions de volumes, particulièrement en 3D pour des directions arbitraires des normales de surface et pour les directions diagonales des vitesses du réseau. Pour des objets fixes, la procédure n'est exécutée qu'une seule fois alors cet exercice n'est pas répété, mais il est tout de même plus complexe que la préparation des conditions aux frontières par interpolation. Cependant, il était important de mentionner la méthode ici car elle est utilisée dans le logiciel commercial PowerFlow.

Dans cette classe de conditions aux frontières on peut aussi inclure les travaux de Verberg et Ladd [66] et de Yu et al. [79]. Dans leurs méthodes, les particules réfléchies par les frontières dépendent du volume des cellules occupé par le solide. Ces méthodes requièrent donc aussi au préalable l'exécution de calculs d'interception de volumes. Ils sont cependant moins élaborés que pour la méthode volumétrique de Chen et al. [75].

2.2.3 Approche immergée

Cette classe de conditions aux frontières a gagné en popularité au cours des dernières années. Elle se distingue des méthodes discutées précédemment parce qu'elle est mieux adaptée aux objets en mouvement et aux frontières déformables. Le principe général des conditions aux frontières immergées consiste à considérer que le fluide occupe aussi l'intérieur d'un objet, et à appliquer des

corrections aux distributions des noeuds voisins de la frontière afin que la vitesse du fluide interpolée sur la frontière soit égale à la vitesse de l'objet au même endroit.

Feng et Michaelides [29] furent les premiers à proposer l'utilisation de cette méthode dans le contexte de la LBM. Ils étudièrent le mouvement de particules dans un fluide et conclurent que, contrairement aux autres méthodes de conditions aux frontières utilisées dans la LBM, la méthode des conditions aux frontières immergées permettait de simuler des problèmes avec particules déformables. Par la suite, la méthode a été réutilisée et adaptée pour des problèmes particuliers par plusieurs auteurs. Zhang et al. [80] ont étudié le mouvement et la déformation de capsules liquides dans des écoulements microfluidiques, Strack et Cook [81] ont entre autres calculé le coefficient de traînée sur une sphère pour des nombres de Reynolds bas, Shu et al. [82] ont calculé la force de traînée exercée par un poisson en mouvement en représentant ce poisson par un profil NACA0012, Shu et Wu [83] ont simulé l'écoulement autour d'un cylindre stationnaire, Wu et al. [84] ont étudié l'écoulement autour d'un cylindre stationnaire et en mouvement, Tian et al. [85] ont simulé le mouvement de filaments élastiques induit par un fluide, Rojas et al. [86] et Seta et al. [87] ont simulé l'écoulement de Couette entre cylindres concentriques et l'écoulement autour d'un cylindre pour des nombres de Reynolds bas, et Nishida et Meichin [88] ont effectué une série de simulations avec un cylindre stationnaire et en mouvement pour montrer que leur variante de la méthode produisait des contours de pressions lisses et continus au travers de la frontière. Hormis les études énumérées précédemment, la méthode de conditions aux frontières immergées a attiré l'intérêt de plusieurs autres auteurs [89–91], y compris quelques études plus récentes [92–98].

2.2.4 Approche versatile

Un grand nombre d'études traitant de l'application de conditions aux frontières dans le contexte de la LBM a été répertorié. Ces conditions peuvent généralement être séparées en deux catégories, selon que le réseau soit délimité par une frontière solide (condition de réflexion) ou qu'il continue au travers de cette frontière (condition immergée). Or, on peut aussi constater que dans la littérature, il n'existe aucun modèle unifié pouvant à la fois traiter des frontières immergées ou non-immersées. Il serait donc intéressant d'explorer l'idée de concevoir un modèle de conditions aux frontières versatile, pouvant traiter précisément et efficacement de frontières courbes ou droites, et qui soient immergées ou non. Une telle approche pourrait par exemple être basée sur la combinaison de réflexions de distributions à l'extérieur et l'intérieur d'un objet immergé. Un

champ de pression continu servirait à déterminer les distributions manquantes post-propagation des deux côtés de la frontière de l'objet.

CHAPITRE 3 SYNTHÈSE DE L'ENSEMBLE DU TRAVAIL

Ce chapitre présente l'ensemble des contributions de cette thèse. Puisque le travail a été divisé en trois objectifs distincts, les résultats sont aussi présentés dans cet ordre. Pour chaque partie du travail, un article scientifique a été rédigé et soumis à une revue. Les trois articles ont été publiés. La version soumise de chacun de ces articles est fournie en annexe.

Les articles ont été rédigés de façon à ce qu'ils contiennent suffisamment de détails pour que d'autres chercheurs puissent reproduire les démarches présentées s'ils le désirent. En ce sens, tous les détails des méthodes et algorithmes ne sont pas répétés. Cette section vise plutôt à faire ressortir de quelle façon les objectifs du travail ont été remplis par les méthodes et résultats présentés à la communauté scientifique.

3.1 Nouvelle méthode de conditions aux frontières

Le présent travail a débuté avec un grand intérêt pour appliquer la LBM à un domaine pratique d'ingénierie. Pour cette raison, le premier objectif de la thèse a consisté à développer l'habileté à créer des programmes de simulation d'écoulements de fluides. De plus, à la suite de la revue de la littérature, un sous-objectif a été le développement d'un nouveau modèle de conditions aux frontières.

Le premier objectif est atteint par l'article suivant [99]:

Pellerin N., Leclaire S., Reggio M. Equilibrium distributions for straight, curved, and immersed boundary conditions in the lattice Boltzmann method. *Comput Fluids* 2014:126-35.

Dans la section 2.2 un portrait de l'éventail des méthodes disponibles de conditions aux frontières applicables à des surfaces solides courbes a été dressé. Cet aspect de la LBM est forcément de première importance dans le contexte de cette thèse puisque l'objectif principal consiste à simuler des écoulements autour de profils d'ailes. On peut classer les méthodes répertoriées selon deux catégories générales, selon qu'elles soient applicables à réseau qui occupe l'intérieur d'un objet ou non. Il a été établi qu'il serait intéressant de disposer d'une méthode versatile pour ces deux situations et c'est ce que la nouvelle méthode proposée de conditions aux frontières accomplit.

3.1.1 Description de la méthode

L'idée de départ est venue après la lecture des travaux de Guo et al. [28] et la constatation que l'usage de leur méthode est répandue dans les publications touchant la LBM. Par exemple, c'est une option disponible dans le logiciel en accès libre Palabos [100]. Tel que mentionné précédemment, cette méthode est basée sur la reconstruction de distributions à l'équilibre sur des nœuds solides au moyen d'une extrapolation de la vitesse, conjointement à une approximation de la partie non-équilibrée des distributions sur ces mêmes nœuds. Ces distributions entrent dans le processus normal de collision-et-propagation et permettent de déterminer les distributions post-propagation manquantes sur les nœuds fluides bordant la frontière. La Fig. 3.1a illustre la méthode de Guo et al. pour une distribution manquante dans la direction 6 d'un réseau d2q9. Les notations utilisées dans l'article original ont été conservées. Dans ce cas où $\Delta=1/2$, la distribution manquante est calculée par les équations suivantes :

$$f_6 = \bar{f}_6^{eq} + \left(1 - \frac{\Delta t}{\Delta t_R}\right) \bar{f}_6^{neq} , \quad (3.1)$$

$$\mathbf{u}_{w1} = \mathbf{u}_f + \frac{1}{\Delta} (\mathbf{u}_b - \mathbf{u}_f) , \quad (3.2)$$

$$\mathbf{u}_{w2} = \mathbf{u}_{ff} + \frac{2}{1+\Delta} (\mathbf{u}_b - \mathbf{u}_{ff}) , \quad (3.3)$$

$$\bar{\mathbf{u}}_w = \Delta \mathbf{u}_{w1} + (1-\Delta) \mathbf{u}_{w2} , \quad (3.4)$$

$$\bar{\rho}_w = \rho_f , \quad (3.5)$$

$$\bar{f}_6^{eq} = \frac{1}{36} \left[\bar{\rho}_w + \rho_0 \left(\frac{\mathbf{c}_6 \cdot \bar{\mathbf{u}}_w}{c_s^2} + \frac{(\mathbf{c}_6 \cdot \bar{\mathbf{u}}_w)^2}{2c_s^4} - \frac{\bar{\mathbf{u}}_w \cdot \bar{\mathbf{u}}_w}{2c_s^2} \right) \right] , \quad (3.6)$$

$$\bar{f}_6^{neq} = \Delta f_{6,f}^{neq} + (1-\Delta) f_{6,ff}^{neq} . \quad (3.7)$$

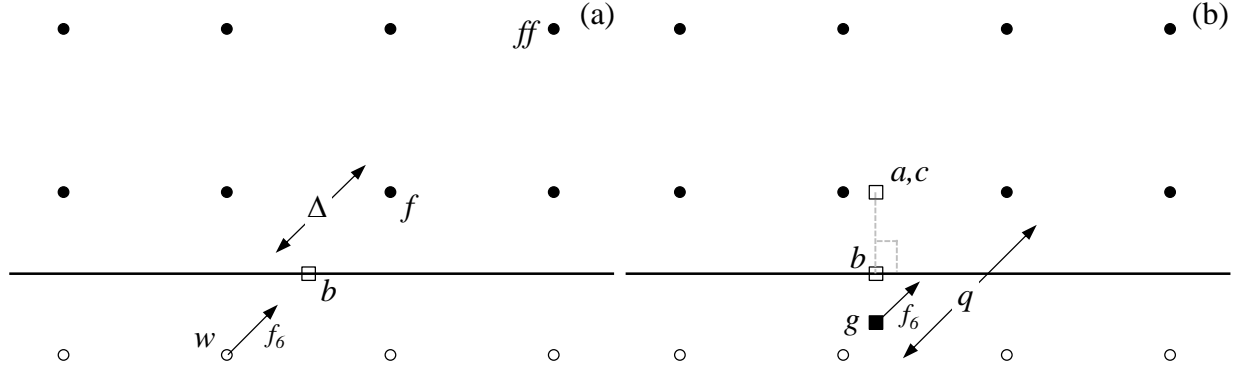


Figure 3.1 : Schémas pour l'application des conditions aux frontières selon a) Guo et al. [28] et b) la nouvelle méthode proposée. Les cercles pleins représentent des nœuds fluides et les cercles vides représentent des nœuds solides.

Cette méthode nécessite de procéder à des extrapolations tant pour les distributions à l'équilibre et non-équilibrées, et de recourir à une étape de collision. Or, la nouvelle méthode proposée vise à vérifier s'il est possible de simplifier cette procédure et de n'utiliser que des distributions à l'équilibre pour la propagation. Comme dans ce cas précis l'article produit ne fournit qu'une explication textuelle du concept de la méthode proposée, son assise mathématique est produite ici. Pour ce faire, il faut recourir à l'Éq. (1.45) obtenue par une expansion Chapman-Enskog. Cette équation est répétée ici:

$$f_i^{neq} = -\Delta t_R \left(\mathbf{c}_i \cdot \nabla + \frac{\partial}{\partial t} \right) f_i^{eq} .$$

Comme cette relation contient une dérivée totale des f_i^{eq} , elle peut être approximée par une forme de différence finie :

$$f_i^{neq}(\mathbf{x}, t) = -\Delta t_R \frac{f_i^{eq}(\mathbf{x}, t) - f_i^{eq}(\mathbf{x} - \mathbf{c}_i \Delta t, t - \Delta t)}{\Delta t} . \quad (3.8)$$

En reformulant l'Éq. (3.8), on obtient

$$f_i^{neq}(\mathbf{x}, t) + \frac{\Delta t_R}{\Delta t} f_i^{eq}(\mathbf{x}, t) = \frac{\Delta t_R}{\Delta t} f_i^{eq}(\mathbf{x} - \mathbf{c}_i \Delta t, t - \Delta t) , \quad (3.9)$$

$$f_i^{neq}(\mathbf{x}, t) + f_i^{eq}(\mathbf{x}, t) + \left(\frac{\Delta t_R}{\Delta t} - 1 \right) f_i^{eq}(\mathbf{x}, t) = \frac{\Delta t_R}{\Delta t} f_i^{eq}(\mathbf{x} - \mathbf{c}_i \Delta t, t - \Delta t) . \quad (3.10)$$

En introduisant la relation $f_i = f_i^{eq} + f_i^{neq}$ dans l'Éq. (3.10), on arrive à :

$$f_i(\mathbf{x}, t) + \left(\frac{\Delta t_R}{\Delta t} - 1 \right) f_i^{eq}(\mathbf{x}, t) = \frac{\Delta t_R}{\Delta t} f_i^{eq}(\mathbf{x} - \mathbf{c}_i \Delta t, t - \Delta t) . \quad (3.11)$$

Maintenant remplaçons Δt par la valeur arbitraire Δt^* en assumant que le processus de collision-et-propagation débute à $\mathbf{x} - \mathbf{c}_i \Delta t^*$ et $t - \Delta t^*$:

$$f_i(\mathbf{x}, t) + \left(\frac{\Delta t_R}{\Delta t^*} - 1 \right) f_i^{eq}(\mathbf{x}, t) = \frac{\Delta t_R}{\Delta t^*} f_i^{eq}(\mathbf{x} - \mathbf{c}_i \Delta t^*, t - \Delta t^*) . \quad (3.12)$$

En choisissant d'utiliser $\Delta t^* = \Delta t_R$, la distribution $f_i(\mathbf{x}, t)$ ne dépend plus que de la distribution à l'équilibre au pas de temps précédent :

$$f_i(\mathbf{x}, t) = f_i^{eq}(\mathbf{x} - \mathbf{c}_i \Delta t_R, t - \Delta t_R) . \quad (3.13)$$

On peut donc interpoler (ou extrapoler) les propriétés macroscopiques à $\mathbf{x} - \mathbf{c}_i \Delta t_R$ et $t - \Delta t_R$ afin de déterminer $f_i(\mathbf{x}, t)$. Or, si on assume que l'écoulement est stationnaire ou quasi-stationnaire à proximité de la frontière, on peut éliminer l'interpolation temporelle en utilisant la relation suivante:

$$f_i(\mathbf{x}, t) = f_i^{eq}(\mathbf{x} - \mathbf{c}_i \Delta t_R, t - \Delta t) . \quad (3.14)$$

En somme, la distribution post-propagation manquante sur un nœud voisin de la frontière peut être approximée en calculant une distribution à l'équilibre basée sur l'interpolation ou l'extrapolation spatiale de la vitesse et de la densité, lesquelles sont effectuées à un nœud fantôme situé à une distance $q = |\mathbf{c}_i| \Delta t_R$ du nœud voisin de la frontière. Cette construction géométrique est illustrée à la Fig. 3.1b, où le nœud fantôme est identifié par la lettre g . De plus, comme le montre l'exemple choisi, la direction normale à la paroi est utilisée pour projeter les quantités macroscopiques sur ce nœud. Aux points a et c , les quantités sont interpolées de façon bilinéaire au moyen des nœuds fluides. Afin de ne pas surcharger le texte, le lecteur est invité à consulter l'article en annexe pour obtenir les détails des procédures d'interpolation et d'extrapolation, le cas échéant. Mentionnons seulement la forme finale de f_6 pour l'exemple choisi :

$$f_6 = \rho_m w_6 \left(1 + \frac{\mathbf{c}_6 \cdot \mathbf{u}_g}{c_s^2} + \frac{(\mathbf{c}_6 \cdot \mathbf{u}_g)^2}{2c_s^4} - \frac{\mathbf{u}_g \cdot \mathbf{u}_g}{2c_s^2} \right). \quad (3.15)$$

Tel qu'indiqué précédemment, cette nouvelle méthode de conditions aux frontières se veut versatile, en étant applicable à des objets immergés ou non. Cette versatilité provient de ρ_m dans l'Éq. (3.15). Dans le cas d'un réseau immergé, ρ_m provient d'une interpolation bilinéaire qui utilise à la fois des nœuds à l'extérieur et l'intérieur d'un objet, ce qui permet la continuité de la densité, ou de la pression, au travers de la frontière. Pour le cas non-immersé, ρ_m est soit approximé par ρ_a ou la densité nominale ρ_0 , selon les conditions décrites dans l'article.

3.1.2 Validation

Afin d'évaluer la méthode proposée, un programme LBM 2D a été développé, lequel utilise un réseau régulier et l'opérateur de collision MRT. Ce choix d'opérateur est nécessaire afin d'assurer la stabilité numérique des calculs. La validation de la méthode est effectuée au moyen de cas classiques de la littérature pour lesquels des solutions analytiques sont connues ou les données d'autres auteurs sont disponibles. De plus, comme la méthode proposée est inspirée par celle de Guo et al. [28], cette dernière est incluse dans le programme pour des fins de comparaison. La méthode de Bouzidi et al. [27] l'est aussi, parce qu'elle est simple, couramment utilisée et basée sur une généralisation du «bounce-back» plutôt que sur l'extrapolation de distributions à l'équilibre. Afin d'alléger la lecture, on réfère désormais à ces méthodes simplement par Guo et Bouzidi.

3.1.2.1 Cavité entraînée

Le premier cas de validation est celui de la cavité entraînée, pour laquelle une vitesse transversale est imposée sur le côté supérieur de domaine, ce qui entraîne la création de tourbillons. Ce cas est classique et les résultats numériques de Ghia et al. [101], obtenus par la résolution des équations de Navier-Stokes sous la forme courant-vorticité, sont souvent utilisés pour des fins de comparaison. Ils le sont aussi dans le présent travail, de même que les résultats numériques de Erturk et al. [102], obtenus au moyen d'une méthode similaire. Une simulation est effectuée pour un nombre de Reynolds de 10^4 et un réseau comportant un espacement similaire à celui du maillage de Ghia et al.. De plus, la méthode proposée de conditions aux frontières est utilisée pour imposer

la vitesse transversale U_0 et des conditions de rebond («bounce-back») sont imposées sur toutes les parois solides. L'objectif de cette configuration consiste à isoler la contribution de la nouvelle méthode pour l'imposition d'une vitesse tangentielle. La même configuration est utilisée pour imposer la vitesse tangentielle avec les méthodes de Guo et Bouzidi.

Pour toutes les conditions aux frontières, l'écoulement obtenu est stationnaire, ce qui était attendu. Les résultats sont comparés au moyen de profils de vitesse normalisée aux plans médians du domaine (Fig. 4, p. 132). Les résultats obtenus avec la nouvelle méthode et celle de Guo sont similaires et concordent mieux avec les résultats de Erturk et al. qu'avec ceux de Ghia et al., bien que les premiers aient été obtenus avec un maillage deux fois plus raffiné. La simulation avec la méthode de Bouzidi produit les bons profils de vitesse mais des amplitudes légèrement supérieures. Ces résultats confirment la validité de la méthode proposée et du programme LBM développé.

3.1.2.2 Écoulement de Couette entre plaques parallèles

Dans l'approche progressive de validation, l'écoulement de Couette entre plaques parallèles est choisi comme deuxième cas. La même configuration de réseau que pour le premier cas est utilisée mais des conditions aux frontières périodiques sont imposées des côtés gauche et droit. L'objectif de la simulation est encore de valider la méthode pour l'imposition d'une vitesse tangentielle mais cette fois-ci pour un écoulement dont on dispose d'une solution analytique. Ce type de solution permet de quantifier l'erreur des résultats obtenus. On veut aussi évaluer la variation de cette erreur en fonction du coefficient de relaxation $\Delta t/\Delta t_R$ et de l'espacement du réseau Δx . Pour ce faire, le nombre de Reynolds est varié entre 0.6 à 30 et le nombre de nœuds couvrant un côté du domaine carré varie entre 4 à 80. Comme dans la configuration de réseau choisie pour ce cas les nœuds de la rangée supérieure se trouvent à exactement $0.5\Delta x$ de la limite supérieure du domaine, l'application de la méthode implique que la vitesse est toujours extrapolée car les nœuds fantômes se trouvent en dehors du domaine. Ce cas est donc intéressant pour évaluer l'impact de la distance d'extrapolation.

Pour les trois méthodes de conditions aux frontières évaluées, l'évolution de l'erreur du champ de vitesse en fonction du nombre de nœuds (Fig. 5, p. 133) montre une décroissance exponentielle légèrement supérieure à 2, ce qui indique que, tout comme les méthodes de Guo et Bouzidi, la méthode proposée produit une exactitude du 2^e ordre. Cet attribut est nécessaire pour que la

méthode puisse être considérée comme une alternative aux méthodes existantes. Pour ce qui est de l'évolution de l'erreur en fonction de $\Delta t_R/\Delta t$, présentée pour le réseau le plus raffiné (Fig. 6, p. 133), les résultats font ressortir trois aspects :

- la méthode de Bouzidi est la moins précise lorsque $\Delta t_R/\Delta t$ est bas,
- la nouvelle méthode est légèrement plus précise que celle de Guo lorsque $\Delta t_R/\Delta t$ est bas,
- la nouvelle méthode devient moins précise que celle de Guo lorsque $\Delta t_R/\Delta t$ dépasse 1, et éventuellement moins précise que celle de Bouzidi lorsque $\Delta t_R/\Delta t$ devient très grand.

Ces observations ne sont pas surprenantes et concordent avec la façon dont la méthode a été construite. En effet, la quantité $\Delta t_R/\Delta t$ est analogue à la distance des nœuds fantômes. Dans la limite où $\Delta t_R/\Delta t$ s'approche de 0.5, le nœud fantôme est tout près de la frontière du domaine et la vitesse extrapolée s'approche de U_0 , donc l'erreur attribuable à l'extrapolation est faible. Au contraire, lorsque $\Delta t_R/\Delta t$ dépasse 1, les nœuds fantômes sont situés plus loin que les nœuds solides utilisés pour la méthode de Guo, ce qui entraîne un accroissement de l'erreur due à l'extrapolation.

3.1.2.3 Écoulement de Couette entre cylindres concentriques

Le troisième cas choisi a pour but d'introduire des frontières courbes tout en disposant encore d'une solution analytique pour l'écoulement, c'est pourquoi l'écoulement de Couette entre cylindres concentriques est retenu. Pour ce dernier, le cylindre intérieur tourne et le cylindre extérieur est fixe. La même configuration de domaine que précédemment est utilisée et les cylindres y sont superposés. Les conditions aux frontières sont imposées de chaque côté des deux cylindres et des conditions de rebond sont imposées sur les quatre côtés du domaine numérique. Un écoulement est donc aussi simulé à l'extérieur du domaine compris entre les deux cylindres mais son champ de vitesse n'est pas utile dans le calcul de l'erreur. Tout comme pour le cas précédent, des simulations sont effectuées en variant l'espacement du réseau et le nombre de Reynolds. Suite aux résultats du deuxième cas, le nombre de Mach de la simulation est aussi été ajusté de façon à ce que $\Delta t_R/\Delta t$

soit toujours près de 0.5, afin d'éviter de trop grandes distances pour les extrapolations de vitesse, ce qui entraîne une baisse de précision de la méthode proposée.

L'analyse de l'évolution de l'erreur en fonction de l'espacement du réseau (Fig. 7, p. 134) confirme que la nouvelle méthode produit une exactitude du 2^e ordre. De plus, pour tous les nombres de Reynolds simulés (Fig. 8, p. 135), l'erreur est inférieure à celle des méthodes de Guo et Bouzidi. Cette observation va de pair avec le choix de limiter $\Delta t_R / \Delta t$. Par ailleurs, la variation de la masse entre les cylindres est mesurée pour toutes les méthodes, incluant la version immergée de la nouvelle méthode. Tel que mentionné précédemment, certains auteurs comme Rohde et al. [63] ont identifié le manque de conservation de masse comme étant une faiblesse du modèle de Bouzidi. Les résultats de la présente simulation (Fig. 9, p. 135) confirment que cette méthode conserve moins bien la masse que la méthode de Guo et que la version non-immersée de la nouvelle méthode. Cependant, on observe une légère variation temporelle de la masse totale entre les cylindres pour la version immergée. Par contre, l'accroissement mesuré de la masse est inférieur d'un ordre de grandeur à celui mesuré pour la méthode de Bouzidi.

3.1.2.4 Cylindre dans un écoulement libre

Le dernier cas de validation est choisi spécifiquement pour confirmer que la version immergée de la méthode produit un champ de densité, ou de pression, continu au travers d'une frontière courbe. Le cas du cylindre dans un écoulement libre est donc retenu, car beaucoup de résultats sont disponibles dans la littérature, incluant pour la LBM. Les dimensions du domaine et les méthodes de conditions aux frontières de ce dernier sont fournies dans l'article.

Des simulations sont effectuées pour des écoulements stationnaires, c'est-à-dire lorsque le nombre de Reynolds est inférieur à environ 46 [103], et pour des écoulements instationnaires produisant des allées de tourbillons de Kármán. Les résultats obtenus, en termes de coefficients aérodynamiques, concordent généralement bien avec les résultats d'autres auteurs. Par exemple, pour $Re=20$ et $Re=40$ (Tableau 2, p. 137), les coefficients de traînée sont proches de ceux obtenus par Izham et al. [104], qui utilisent eux aussi la LBM sur un réseau régulier, conjointement avec la méthode de Guo pour imposer les conditions aux frontières sur le cylindre. Pour les écoulements instationnaires à $Re=100$ et $Re=200$, on mesure le nombre de Strouhal, un indicateur adimensionnel de la fréquence des tourbillons générés dans le sillage. Les résultats obtenus (Tableau 3, p. 137)

concordent encore une fois bien avec ceux de Izham et al. [104] et aussi avec celui de Chen et al. [74] pour $Re=200$, qui utilisent eux aussi la LBM sur un réseau régulier.

Par ailleurs, une mesure du nombre de Strouhal pour des simulations effectuées à divers nombres de Reynolds permet de déterminer que le régime instationnaire débute entre $Re=46$ et $Re=46.5$ (Fig. 11, p. 138), ce qui correspond au nombre critique de Reynolds cité par Higura et Succi [103].

Deux autres constatations intéressantes à-propos de la nouvelle méthode proposée ressortent de ce cas de validation. Premièrement, le graphique des contours du champ de densité présenté dans l'article (Fig. 12, p. 139) confirme que cette densité varie de façon assez lisse au travers de la frontière du cylindre, ce qui était l'objectif visé. Deuxièmement, le profil de coefficient de pression sur la circonférence du cylindre (Fig. 13, p. 139) concorde mieux avec le résultat comparatif choisi que ceux obtenus avec les méthodes de Guo et Bouzidi. Ces méthodes génèrent plus d'ondulations dans les zones où la normale de la surface du cylindre est la moins alignée avec les directions orthogonales du réseau. Cette observation indique que le choix de positionnement des points d'interpolations le long de la normale à la surface est probablement bénéfique.

3.1.3 Conclusions

Les travaux réalisés dans le cadre de la rédaction du premier article permettent d'atteindre le premier objectif, soit de développer les aptitudes à programmer la LBM et de proposer une nouvelle méthode de conditions aux frontières. Cette méthode a été validée au moyen de quatre cas, lesquels permettent d'établir qu'elle démontre une exactitude du 2^e ordre et qu'elle peut effectivement être utilisée dans les contextes où un objet est immergé ou non sur un réseau. Cependant, les analyses montrent aussi que la précision de la méthode est optimale lorsque $\Delta t_R / \Delta t$ est près de 0.5, ce qui correspond à la limite de sur-relaxation de l'opérateur de collision. Ainsi, la méthode apparaît plus appropriée pour les écoulements à hauts Re . Pour cette raison, la méthode est aussi utilisée pour la deuxième partie du travail, présentée à la section suivante.

3.2 Écoulements turbulents sur profils d'ailes avec réseaux multi-domaines

La deuxième partie du travail aborde le thème principal de cette thèse, c'est-à-dire la simulation d'écoulements turbulents sur des profils d'ailes. L'approche choisie pour effectuer ces simulations provient de deux lacunes identifiées par la revue de littérature. Premièrement, pour les écoulements turbulents considérés, le nombre de Reynolds est nécessairement élevé et en conséquence le coefficient de relaxation $\Delta t/\Delta t_r$ des simulations approche la limite de 2. Or, bien que dans cette situation l'opérateur de collision BGK soit reconnu pour introduire des instabilités numériques, certains auteurs l'ayant utilisé n'en font pas mention et n'indiquent pas non plus s'ils ont utilisé des stratégies particulières pour assurer la stabilité. L'opérateur de collision MRT, qui augmente la stabilité, a aussi été utilisé [55,56]. Pour la première de ces deux études, qui utilise un réseau multi-domaines, les résultats sont discutables car l'auteur obtient un écoulement détaché à un angle d'attaque peu élevé. Pour la deuxième étude, une méthode LBM par interpolation est utilisée. La stabilité des opérateurs BGK et MRT dans le contexte d'un écoulement à haut Re pour lequel on utilise un modèle de turbulence et une configuration de réseau multi-domaines n'est donc pas vérifiée adéquatement. La deuxième lacune identifiée concerne le modèle de turbulence. Le modèle Spalart-Allmaras est un bon candidat car il implique moins d'équations à résoudre que le modèle $k-\varepsilon$ par exemple. Cependant, la seule étude répertoriée utilisant ce modèle avec un réseau multi-domaines, jugé approprié pour des objets aux formes arbitraires, est incomplète. Cette configuration, c'est-à-dire multi-domaines et S-A, nécessite donc d'être explorée.

Lors de simulations préliminaires effectuées avec les opérateurs BGK et MRT pour un cylindre soumis à un nombre de Reynolds très élevé, des instabilités numériques sont survenues. Cette constatation a mené à identifier une méthode alternative offrant plus de robustesse. L'opérateur en cascade a été sélectionné.

Le deuxième objectif de la thèse consistait donc à développer un programme LBM utilisant une configuration multi-domaines, l'opérateur de collision en cascade et le modèle de turbulence Spalart-Allmaras. Cette configuration n'est pas proposée dans la littérature, et l'emploi combiné du réseau multi-domaines et de l'opérateur en cascade n'a été proposé [58] qu'après les présents travaux.

Le deuxième objectif est atteint par l'article suivant [105]:

Pellerin N., Leclaire S., Reggio M. An implementation of the Spalart-Allmaras turbulence model in a multi-domain lattice Boltzmann method for solving turbulent airfoil flows. *Comput Math with Appl* 2015;70:3001-18.

3.2.1 Développement du programme

Le programme est construit au moyen de quatre composantes principales, soit un réseau de type multi-domaines, l'opérateur de collision en cascade, le modèle de turbulence S-A et le modèle de conditions aux frontières courbes proposé dans le premier article. L'implémentation de chacune de ces composantes est décrite dans le deuxième article. Or, afin de donner au lecteur une bonne perspective globale du programme, il est important d'expliquer le choix de ces composantes et de réexaminer leurs caractéristiques.

3.2.1.1 Réseau multi-domaines

Tout d'abord, il faut rappeler que le choix de l'approche du réseau multi-domaines est fondé sur deux critères :

- la nécessité d'obtenir un espacement de réseau très petit à proximité d'un objet soumis à un écoulement libre à haut Re , et ce tout en limitant le nombre total de nœuds du réseau,
- la nécessité que le raffinement du réseau puisse être localisé dans le cas où l'objet soumis à l'écoulement aurait une forme irrégulière, comme par exemple un profil d'aile sur lequel de la glace de type « glaze » aurait formé des cornes sur le bord d'attaque.

Selon ces critères, une configuration qui emploierait seulement un emboîtement de réseaux rectangulaires ne serait pas satisfaisante. En effet, le réseau le plus fin, contenant l'objet, serait mal adapté et fournirait une haute densité de nœuds dans des régions où c'est moins nécessaire, ce qui nécessiterait du même coup un usage excessif de mémoire d'ordinateur. Une bonne illustration de cette situation serait un profil d'aile à un angle d'attaque de 45° . Il faut donc que les réseaux épousent mieux la forme de l'objet afin d'assurer un raffinement surtout à proximité de ce dernier.

Dans la littérature LBM, il existe plusieurs méthodes de transfert d'information entre les réseaux d'une configuration multi-domaines. Elles peuvent être divisées en deux catégories, soit les méthodes volumétrique et par interpolation. Dans le premier cas, selon des méthodes proposées par Chen et al. [53] et Rohde et al. [106], les réseaux voisins, comportant un espacement différent, sont positionnés de telle sorte qu'une interface de cellules existe. Par exemple, en 2D, ces cellules forment un carré de dimension Δx , centré autour d'un nœud. Ainsi, pour une interface où le facteur de raffinement est 2, quatre cellules du réseau le plus fin sont superposées à celle de l'autre réseau. Le transfert d'information est effectué après les étapes de collisions, alors que les distributions f_i du réseau le moins raffiné sont divisées puis transférées au réseau fin, et que réciproquement les distributions du réseau fin sont regroupées afin d'être transférées à l'autre réseau. Cette procédure, que nous avons simplifiée en omettant des détails techniques, assure une conservation de la masse et de la quantité de mouvement. Pour la deuxième catégorie de méthodes, introduite par Filippova et Hänel [69], l'interface des réseaux voisins est configurée différemment. En effet, les réseaux se chevauchent mais les nœuds sont alignés plutôt que les cellules. De plus comme chaque réseau a un coefficient de relaxation $\Delta t / \Delta t_R$ différent, le transfert d'information est fait de telle sorte que, sur les frontières des réseaux, les distributions à l'équilibre f_i^{eq} soient identiques et les distributions non-équilibrées respectent une proportionnalité qui elle dépend des coefficients de relaxation. L'information est transférée vers le réseau le plus raffiné par interpolation unidimensionnelle sur l'autre réseau, et réciproquement par moyenne pondérée locale ou transfert direct puisqu'il y a un nœud fin sous chaque nœud de la frontière du réseau moins raffiné. Il faut noter que cette méthode par interpolation a aussi été adaptée par Eitel-Amor et al. [107] pour un chevauchement volumétrique aux interfaces combiné à une interpolation bidimensionnelle. Pour le présent travail, c'est la méthode par interpolation qui est retenue, telle que proposée récemment par Lagrava et al. [34], parce qu'ils indiquent que leur implémentation de la méthode permet d'en améliorer la stabilité numérique pour les écoulements à hauts Re, ce qui est d'intérêt primordial pour cette thèse. Ce choix est aussi basé sur les cas de validation qu'ils ont choisis. En effet, bien que Chen et al. [53] indiquent que les méthodes par interpolation ne garantissent pas la conservation des propriétés macroscopiques, Lagrava et al. obtiennent une continuité de la densité au travers d'une interface placée perpendiculairement à un écoulement de Poiseuille. Pour ce faire, ils indiquent qu'une interpolation cubique est nécessaire. De plus, pour un cas de propagation de dipôle à haut Re, où des gradients de vitesse très élevés traversent les interfaces des réseaux, et que

des sous-tourbillons sont formés au contact d’une paroi, ils obtiennent une continuité des champs macroscopiques au travers des interfaces, grâce à l’application d’un filtre pour transférer l’information des réseaux fins vers leur voisin moins raffiné.

Par contre, il est important de mentionner que la méthode de Lagrava et al. [34] est applicable conjointement avec l’opérateur de collision BGK et que ce n’est pas celui que nous utilisons. Nous devons donc vérifier si cette combinaison introduit des discontinuités. Par ailleurs, il existe une méthode de transfert de distributions par interpolation développée pour l’opérateur de collision en cascade [108] mais elle n’est pas utilisée pour le présent travail en raison de la lourdeur des calculs qu’elle implique à chaque pas de temps, incluant la résolution d’un système linéaire d’équations pour chaque nœud où de l’information est transférée.

3.2.1.2 Opérateur de collision

Le choix de l’opérateur de collision est fondé sur des essais préliminaires effectués avec les opérateurs BGK et MRT et un réseau multi-domaines basé sur la méthode de Lagrava et al [34]. Un cylindre a été soumis à un écoulement avec un nombre de Reynolds suffisamment élevé pour générer des coefficients de relaxation proches de la limite de 2 pour tous les niveaux de réseaux. Dans le premier cas, des instabilités numériques ont interrompu l’exécution des calculs assez rapidement. Dans le deuxième cas, l’exécution des calculs a perduré plus longtemps mais des instabilités ont fini par produire des régions de vitesse très élevée qui ont fait diverger la solution. Pour ces raisons, une alternative à ces opérateurs de collision est nécessaire. Tel qu’il est mentionné à la section 1.1.5.2, il existe d’autres options permettant d’augmenter la stabilité. D’emblée la méthode de Ansumali et Karlin [31], communément appelée LBM entropique ou ELBM, n’est pas retenue pour le présent travail simplement parce qu’elle requiert la résolution continue d’équations pour chaque nœud du réseau. Cette caractéristique de la méthode représente un inconvénient non-négligeable pour un réseau qui doit comporter un nombre élevé de nœuds. La méthode de régularisation proposée par Lätt [24], parfois désignée par l’acronyme RLBM, est quant à elle beaucoup moins coûteuse à utiliser et constitue une option intéressante. Or, l’essai préliminaire auquel ont été soumis les modèles BGK et MRT a aussi été utilisé pour vérifier le modèle par régularisation dans le contexte du programme développé pour la thèse. Il s’est avéré que les calculs ont produit des anomalies dans le champ de vitesse plus rapidement que pour le modèle MRT. Suite à ce constat, des essais ont aussi été effectués sur le modèle d’opérateur de

collision en cascade [32,33]. Les calculs sont demeurés stables, indépendamment du nombre de Reynolds choisi. Pour cette raison, c'est le modèle retenu pour le développement du programme. Une brève description du concept modèle est fournie à la section 1.1.5.2 et elle ne sera pas répétée ici. Cependant, il est à propos de réitérer les raisons qui lui confèrent ses propriétés de stabilité. Tout d'abord, le traitement séquentiel des moments des distributions permet de découpler les échelles dynamiques. Ainsi, les instabilités qui émanent généralement des hautes fréquences du mouvement du fluide, c'est-à-dire des longueurs d'ondes proches de l'espacement du réseau, perturbent moins l'écoulement car les moments des distributions correspondant aux modes hydrodynamiques tels la diffusion de la quantité de mouvement par la viscosité sont relaxés avant les moments d'ordres supérieurs. Deuxièmement, les moments centraux à l'équilibre, utilisés comme «attracteurs» des collisions, sont indépendants de la vitesse, ce qui, selon Geier et al. [33], augmente l'invariance galiléenne. De plus, comme ils ne sont pas obtenus par une expansion de Taylor en série polynomiale autour de la vitesse macroscopique nulle, tel que pour l'Éq. (1.38), le processus de collision est plus stable que pour les opérateurs utilisant cette approximation quand la vitesse locale s'éloigne de l'origine. Les détails complets de la dérivation du modèle sont disponibles dans la thèse de Geier [32] mais l'expression des moments centraux à l'équilibre, basée sur l'Éq. (1.36), est produite ici pour bien illustrer le concept :

$$\kappa_{x^m y^n}^{eq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (c_x - u_x)^m (c_y - u_y)^n \rho \left(\frac{1}{2\pi c_s^2} \right) e^{\left(\frac{-(c-u)(c-u)}{2c_s^2} \right)} dx dy . \quad (3.16)$$

Cette intégration produit des moments d'équilibre non-nuls pour $\kappa_{x^0 y^0}^{eq} = \rho$, $\kappa_{x^2 y^0}^{eq} = \kappa_{x^0 y^2}^{eq} = \rho c_s^2$ et $\kappa_{x^2 y^2}^{eq} = \rho c_s^4$, tous indépendants de la vitesse.

3.2.1.3 Modèle de turbulence

Le programme est développé pour simuler un type d'écoulement qui émane de l'idée originale de ce travail, c'est-à-dire d'utiliser la LBM pour simuler la formation de glace sur des ailes d'avions. Or, les conditions de formation de glace sont généralement rencontrées à des altitudes supérieures à 10000 pieds par des aéronefs qui volent à des centaines de kilomètres par heure. Selon les propriétés de l'air à de telles attitudes et la dimension des ailes des avions, le nombre de Reynolds des écoulements est de l'ordre de 10^6 , typiquement dans le régime turbulent. Même des aéronefs tels les drones militaires, qui volent moins vite que des avions de ligne et qui ont de plus petites

ailes, sont assujettis à des écoulements turbulents. Le programme doit donc pouvoir simuler ce type d'écoulement. Par ailleurs, des approches simplifiées telles le recours aux écoulements potentiels ont déjà été utilisées dans le module aérodynamique de certains logiciels de simulation de formation de glace. Cependant, dans l'optique d'atteindre un niveau supérieur de fidélité des simulations, le champ de vitesse près d'un profil doit aussi être calculé avec précision, c'est pourquoi les effets de la viscosité et de la turbulence doivent être inclus. De plus, si on omet le contexte de formation de glace, l'étude des écoulements turbulents sur profils d'ailes au moyen de la LBM demeure un sujet qui nécessite d'être abordé car il y a des lacunes dans la littérature.

Le programme est développé pour simuler des écoulements autour de profils d'ailes, donc le domaine numérique est bidimensionnel. Or, dans ce contexte, comme la turbulence est un phénomène à la base tridimensionnel, toutes les échelles de turbulence doivent être modélisées. Il existe plusieurs modèles pour traiter de la turbulence de cette façon, que l'on peut diviser en trois catégories : algébrique, à une équation et à deux équations ou plus. Pour le présent travail, le choix s'arrête sur un modèle à une équation, soit Spalart-Allmaras [109]. Ce choix est basé sur plusieurs éléments. Premièrement, comme l'introduction d'un modèle de turbulence déroge de la LBM pure en y ajoutant la résolution d'une ou plusieurs équations différentielles sur le même réseau, l'utilisation d'un modèle à une seule équation limite l'effort de calcul additionnel requis. Deuxièmement, tel qu'indiqué à la section 1.2.2, l'utilisation du modèle S-A conjointement avec un réseau en configuration multi-domaines n'a pas été explorée adéquatement. Troisièmement, Lockard et al. [110] rapportent que ce modèle est ajusté pour les écoulements aérodynamiques, qu'il est robuste et qu'il est applicable à des écoulements sur des géométries plus complexes. Finalement, comme le sujet principal de ce travail n'est pas le modèle de turbulence en soi, le choix du modèle S-A a un côté pragmatique, c'est-à-dire que c'est un modèle fréquemment utilisé pour le type d'écoulements considérés ici et il existe des résultats dans la littérature pouvant être utilisés pour des fins de comparaison.

Les équations et les constantes du modèle sont produites dans l'article et ne sont pas répétées ici, de même que l'algorithme complet pour le domaine multi-domaines. Il est par contre pertinent de rappeler qu'une méthode de différence-finie est choisie pour résoudre l'équation différentielle du modèle parce que c'est l'approche qui se prête le plus naturellement à un réseau régulier.

3.2.1.4 Conditions aux frontières du profil d'aile

Le dernier aspect de la construction du programme concerne l'application de conditions aux frontières de glissement nul sur le profil. Puisque que pour remplir le premier objectif de la thèse une nouvelle méthode de conditions aux frontières a été développée et que les résultats de sa validation ont permis de conclure qu'elle était appropriée pour les écoulements à hauts Re , c'est cette méthode qui est retenue pour le programme. Cependant, une méthode simplifiée n'utilisant qu'un seul point d'interpolation dans le fluide est introduite (Fig. 3, p. 155). Cette modification est apportée afin de simplifier la programmation et aussi de positionner tous les points d'interpolation à la même distance de la frontière dans l'optique future d'introduire aussi des fonctions de paroi. Cependant, ce dernier aspect ne fait partie du présent travail car l'objectif consiste à appliquer un modèle de turbulence dans sa formulation de base et les données comparatives obtenues par d'autres études n'utilisent pas non plus ce type d'approximation près des parois. C'est d'ailleurs pour cette raison que les nombres de Reynolds retenus pour la validation du programme se situent dans la limite inférieure des écoulements turbulents, afin de limiter le raffinement nécessaire pour bien capturer la couche limite.

3.2.2 Validation

Le premier cas de validation sélectionné est celui d'un écoulement sur un profil NACA 0012 à $Re=5 \times 10^5$. Ce cas est choisi parce que le nombre de Reynolds est relativement bas et que des résultats d'autres études sont disponibles dans la littérature. En effet, Li et al. [56] utilisent une méthode LBM par interpolation couplée au modèle S-A, donc il est très intéressant de comparer cette approche avec celle du présent travail. De plus, Lockard et al. [110] ont produit des résultats avec le logiciel établi de la NASA CFL3D, lequel emploie une formulation en volumes finis des équations de Navier-Stokes sur un maillage de type C-grid. Ces résultats ont aussi été générés avec le modèle de turbulence S-A.

Le domaine numérique consiste en un carré dont les côtés mesurent 200m. Le profil a une corde de 1m et son bord d'attaque est positionné au centre du domaine. Les réseaux de la configuration multi-domaines sont construits afin de produire des couches successives dont l'épaisseur est généralement de 12 nœuds. Un facteur de 2 est utilisé pour le raffinement de tous les réseaux voisins. Dans l'article, cet arrangement est illustré pour un raffinement maximum de $1/512^\circ$ de corde (Fig. 2, p. 152). Or, pour les simulations, un raffinement maximum de $1/8192^\circ$ de corde est

utilisé afin d'être comparable à celui de CFL3D. Les côtés du domaine numérique étant couverts par 201 nœuds, 14 niveaux de réseaux sont nécessaires pour obtenir le raffinement désiré sur le profil. Des simulations sont effectuées pour des angles d'attaque de 0° , 3° , 7° et 12° et le nombre total de nœuds est d'environ 550000 dans chacun des cas. Il faut noter que l'espacement du réseau est le même sur tout le périmètre du profil et qu'un raffinement plus localisé n'est pas utilisé. Ce choix sera expliqué plus loin. L'algorithme implémenté dans le programme pour procéder aux calculs sur chaque réseau et transférer l'information entre ces derniers est fourni dans l'article.

Les résultats obtenus sont comparés sur la base des coefficients aérodynamiques, des coefficients de pression et des profils de vitesse tangentielle. Dans le premier cas les forces de traînée et de portance exercées par le fluide sur le profil sont calculées simplement par la méthode de variation de quantité de mouvement [111], laquelle utilise la différence entre les distributions propagées vers la frontière solide et celles en provenance de la frontière, donc dans le cas présent les distributions émanant des nœuds fantômes. Les résultats sont disponibles dans l'article et reproduits ici au Tableau 3.1. Les coefficients de portance concordent bien avec ceux de CFL3D, à l'exception de celui obtenu pour l'angle d'attaque le plus élevé, qui montre un écart de -1.6%. Pour ce qui est des coefficients de traînée, l'écart est un peu plus grand. Le présent programme la sous-estime par 24 points en moyenne.

Tableau 3.1 : Coefficients aérodynamiques obtenus au moyen du programme multi-domaines.

Angle [°]	C_L		C_D	
	CFL3D	Présent	CFL3D	Présent
0	0.0000	0.0000	0.0128	0.0100
3	0.3237	0.3257	0.0130	0.0104
7	0.7449	0.7457	0.0157	0.0128
12	1.1809	1.1623	0.0275	0.0263

Afin de vérifier l'effet du raffinement du réseau sur les coefficients aérodynamiques, des simulations sont aussi conduites pour des espacements minimum de réseau variant entre $1/4096^\circ$ et $1/512^\circ$ de corde, tel que présenté dans l'article (Fig. 7, p. 160). Les courbes montrent que l'espacement de $1/8192^\circ$ de corde est suffisant pour les angles d'attaques de 0° à 7° . Pour l'angle de 12° , on ne peut pas tirer la même conclusion car les simulations utilisant moins de raffinement produisent des écoulements détachés. Pour cette raison, une simulation est aussi exécutée pour un raffinement de $1/16384^\circ$ de corde et le coefficient de portance généré est de 1.1740, donc plus près

du résultat de CFL3D. Ce résultat indique que, même pour un nombre de Reynolds relativement peu élevé, lorsque l'angle d'attaque s'approche du point de décrochage aérodynamique, une simulation numérique adéquate nécessite un réseau très raffiné.

Le deuxième élément de comparaison est celui des coefficients de pression. Dans l'article, ils sont présentés pour les angles d'attaque de 0° , 7° et 12° et comparés avec ceux de CFL3D. Les résultats de Li et al. [56] disponibles pour 0° et 7° ne sont pas reproduits afin de ne pas surcharger les graphiques. À 0° et 7° , les résultats concordent bien et les courbes sont pratiquement superposées (Figs. 8-9, p. 160-161). À 12° , il y a une différence près du bord d'attaque (Fig. 10, p. 161). La fossette observée correspond en fait au point où la couche limite passe de laminaire à turbulente. Pour les résultats de CFL3D, cette fossette est localisée plus près du bord d'attaque et son amplitude est plus petite, ce qui indique que la transition se fait plus tôt. On constate que la simulation effectuée pour le raffinement de $1/16384^\circ$ de corde atténue cette fossette et la rapproche du bord d'attaque, mettant encore en évidence la nécessité de raffiner davantage, au moins dans cette région du profil. Près du bord d'attaque du profil, le gradient de vitesse tangentielle à la paroi en direction normale de cette dernière est très élevé et requiert un espacement de réseau très fin pour le capturer adéquatement. Tel qu'indiqué dans l'article, une raison pour laquelle la solution à 12° diffère légèrement de celle de CFL3D réside dans l'application des conditions aux frontières. Dans le présent programme, la vitesse est interpolée à $1.5 \Delta x$ de la paroi alors que pour la méthode en volumes finis de CFL3D, la vitesse utilisée pour imposer la condition de glissement nul est probablement située au centre d'une cellule, soit à approximativement 0.5Δ de la paroi, où Δ est la distance normale entre la paroi et la première rangée de nœuds. Pour $\Delta \approx \Delta x$, cette dernière approche peut donc mieux capturer le gradient pour l'application de la condition aux frontières.

Les profils de vitesse constituent le troisième élément de comparaison, pour lequel les résultats de Li et al. [56] sont aussi présentés. Ces profils montrent la vitesse normalisée tangentielle à l'axe du profil en fonction de la distance normale à cet axe (Figs. 11-13, p. 162-164). On peut en tirer trois observations :

- les profils obtenus au moyen du programme multi-domaines ont généralement la même forme que ceux de CFL3D mais sont un peu plus «remplis», c'est-à-dire que la vitesse est un peu plus grande dans un intervalle de distance intermédiaire à la paroi,

- les profils de Li et al. diffèrent de ceux de CFL3D, particulièrement près du bord de fuite où la différence est marquée, tant pour l'amplitude que la forme,
- pour l'angle d'attaque de 12° , le profil du présent programme près du bord d'attaque contient moins de quantité de mouvement que celui de CFL3D.

Cette dernière observation met encore à l'avant-plan la question du raffinement, particulièrement près du bord d'attaque.

Le programme a aussi été utilisé pour simuler les écoulements sur des profils S1223 et E387 afin de présenter un deuxième cas de validation. Les formes géométriques de ces profils sont illustrées dans l'article. Ils sont choisis parce qu'ils ne sont pas symétriques et ont été conçus pour obtenir une portance élevée à des nombres de Reynolds moins élevés. De plus, des données expérimentales sont disponibles dans la littérature. Des simulations sont effectuées à $Re=2 \times 10^5$ et des angles d'attaque couvrant la gamme de points expérimentaux disponibles. Les réseaux multi-domaines sont construits de la même façon que pour le profil NACA 0012 mais avec un espacement de $1/4096^\circ$ de corde près du profil parce que le nombre de Reynolds est moins élevé. Le nombre total de nœuds est d'environ 315000 pour toutes les simulations.

Les résultats sont comparés dans l'article en termes de coefficients de portance et de traînée (Figs. 14-15, p. 164-165). Pour le profil E387, on observe une bonne concordance avec les données expérimentales. Par contre, le coefficient de portance maximal est un peu surestimé et le décrochage aérodynamique se produit à un angle d'attaque inférieur à celui obtenu expérimentalement. Quant au profil S1223, les résultats concordent le mieux entre 0° et 10° . Cependant, le décrochage aérodynamique est observé plus tôt et son impact sur la réduction de portance est plus fort. De plus, pour les angles d'attaques négatifs, les courbes de portance montrent une différence marquée, alors que la réduction de portance est observée plus rapidement avec le présent programme à mesure que l'angle est réduit. Pour ces angles, l'écoulement sous le profil commence à se détacher et il est possible que le programme ne capture pas correctement cette région détachée. Un raffinement de réseau insuffisant peut encore être identifié comme explication possible de ces différences. Il faut aussi noter que les résultats pour le profil E387 sont meilleurs que pour le profil S1223 parce que ce dernier est beaucoup plus cambré et son épaisseur varie plus.

3.2.3 Conclusions

Les travaux réalisés afin de rédiger le deuxième article permettent d'atteindre le deuxième objectif. Un programme a été développé pour simuler des écoulements aérodynamiques au moyen de la LBM appliquée à un réseau de type multi-domaines. Il a été conçu pour pouvoir utiliser un nombre arbitraire de niveaux de réseaux, pouvant ainsi générer un réseau très fin près d'un profil d'aile soumis à un écoulement libre. De plus, le programme est stable numériquement grâce à l'utilisation de l'opérateur de collision en cascade. Il fait aussi usage du modèle de conditions aux frontières développé dans le cadre de la rédaction du premier article et d'un modèle de turbulence applicable au type d'écoulements étudiés. Le choix du modèle de turbulence et de l'approche pour en résoudre l'équation de transport de viscosité est validé par l'obtention d'écoulements stationnaires dont les profils de vitesse correspondent à ceux obtenus pour le même modèle mais avec une approche numérique différente.

De plus, l'utilisation combinée de l'opérateur de collision en cascade et de la méthode multi-domaines de Lagrava et al. [34] développée pour l'opérateur BGK n'a pas causé de problèmes apparents. Les champs macroscopiques sont continus au travers des niveaux de maillage.

Par contre, les résultats montrent quelques limitations du programme. Tout d'abord, l'exécution des calculs jusqu'à convergence des coefficients aérodynamiques a nécessité 10 jours pour le profil NACA 0012 à 12° et entre 4 et 5 jours pour les profils S1223 et E387. Les calculs ont été effectués sur un processeur Intel i7 3.6 GHz couplé à 16 GO de mémoire vive, au moyen d'un programme développé dans l'environnement Matlab, choisi pour la facilité avec laquelle il permet d'effectuer rapidement des modifications et parce qu'il permet de vectoriser les opérations, ce qui simplifie l'écriture du code et accélère les calculs. Cependant, le passage à un langage de programmation comme C++ appuyé par un transfert efficace des calculs sur GPU pourrait certainement permettre d'obtenir des gains de temps d'exécution. Ceci étant dit, la durée des calculs n'est pas seulement attribuable à la vitesse d'exécution du programme mais aussi au nombre élevé d'itérations requises. Il serait donc intéressant d'explorer l'implémentation et l'impact de méthodes d'accélération de convergence.

Une autre limitation du programme dans sa forme actuelle est le raffinement uniforme sur le périmètre du profil. En effet, il serait avantageux par exemple d'utiliser un espacement de réseau plus fin sur le bord d'attaque que sur le reste du profil. Des essais en ce sens ont été conduits mais

ont révélé deux problèmes. Tout d'abord, le traitement du calcul des forces par la méthode de transfert de quantité de mouvement est difficile à effectuer de façon précise à la jonction de deux niveaux de réseaux. Deuxièmement, la plus grande difficulté rencontrée est une discontinuité des champs macroscopiques au passage d'un réseau fin vers son voisin moins raffiné, lorsque la direction de l'écoulement est dans cette direction. Les gradients de vitesse tangentielle à la paroi sont grands à proximité de cette dernière et l'interpolation effectuée pour le transfert d'information introduit des fluctuations anormales de vitesse et de densité. Plus de recherches seraient nécessaires pour corriger ce problème. D'ailleurs, la section suivante de la thèse traite d'une approche permettant un raffinement localisé et directionnel du réseau dans le contexte de la LBM.

3.3 Écoulements turbulents sur profils d'ailes avec réseaux non-structurés

Pour la troisième partie du travail, le thème principal de la thèse est encore abordé mais d'un angle différent de la deuxième partie. En effet, dans la section précédente, le développement d'un programme de simulation d'écoulements turbulents sur profils d'ailes basé sur une configuration de réseau multi-domaines a été décrit et validé. Or, cette approche, du moins dans son implémentation pour la présente thèse, ne permet pas de raffiner le réseau localement sur le périmètre du profil. De plus, un espacement de réseau plus fin près du bord d'attaque, particulièrement dans le cas du profil NACA 0012, s'est montré nécessaire pour améliorer la précision des résultats pour les angles d'attaques élevés. Pour cette raison, le développement d'un programme basé sur un réseau non-structuré est maintenant considéré. Ce type de réseau permet un raffinement directionnel, ce qui pourrait permettre d'améliorer les résultats tout en limitant le nombre de nœuds utilisés. De plus, il permet aussi de s'adapter à des formes géométriques arbitraires, ce qui répond à un des critères de base considérés dans cette thèse pour le développement d'un programme. Par contre, il est admis d'emblée que générer ce type de réseau requiert plus d'efforts que la construction de réseaux réguliers imbriqués.

Le troisième objectif de la thèse consistait donc à développer un programme LBM utilisant une configuration non-structurée pour le réseau, et à l'utiliser pour résoudre des écoulements turbulents sur profils d'ailes. Cette approche n'est pas proposée dans la littérature.

Le troisième objectif est atteint par l'article suivant [112]:

Pellerin N., Leclaire S., Reggio M. Solving incompressible fluid flows on unstructured meshes with the lattice Boltzmann flux solver. Eng Appl Comput Fluid Mech 2017;11:310–27.

3.3.1 Développement du programme

L'utilisation d'un réseau non-structuré déroge de la définition de base de la LBM. En effet, une distribution émanant d'un nœud à une vitesse c_i et se propageant pendant une période Δt n'arrive pas exactement sur l'emplacement d'un nœud voisin. Cette situation est aussi présente dans les

approches de LBM par interpolation utilisant des réseaux orthogonaux déformés. Dans ce cas, on s'assure qu'une distribution propagée atteigne un nœud en interpolant sa valeur à un point de départ situé à $-\mathbf{c}_i \Delta t$. Une transformation de coordonnées permet de recourir à une interpolation spatiale du 2^e ordre. Cependant, dans le cas d'un réseau non-structuré, on ne peut pas transformer les coordonnées en un système orthogonal équivalent afin de simplifier l'interpolation et il est difficile d'établir le schéma d'interpolation spatiale optimal en fonction des nœuds se trouvant à proximité. De plus, il est connu que l'utilisation d'une méthode d'interpolation dans la LBM peut amener de la dissipation numérique [50]. D'ailleurs, des essais réalisés dans le cadre de cette thèse ont été non concluants pour des réseaux où la grandeur des cellules variait de façon significative dans l'espace. Les solutions obtenues pour des écoulements typiques n'étaient pas représentatives des résultats attendus, d'une telle façon que les champs macroscopiques semblaient être assujettis à une viscosité non uniforme. La distance relative locale entre $|\mathbf{c}_i \Delta t|$ et la dimension des cellules est en cause. En effet, afin que l'évolution temporelle de la solution ait une signification physique, la distance de propagation doit être la même pour tous les nœuds, indépendamment de l'espacement local entre les nœuds du réseau non-structuré. Or, pour une approximation spatiale polynomiale du 2^e ordre centrée sur le nœud qui reçoit les distributions propagées, plus la distance relative du point d'interpolation est petite, plus la valeur interpolée correspondra à la partie linéaire du polynôme. Cette linéarisation introduit donc de la dissipation numérique. En raison de ces difficultés, une autre approche devient nécessaire.

Tel que mentionné à la section 1.1.5.3, dans la littérature LBM deux options existent pour utiliser des réseaux non-structurés. La première, la LBM par volumes finis ou FVLBM, a été proposée par Peng et al. [113,114]. Elle est basée sur une formulation intégrale de l'équation de Boltzmann à vitesses finies, pour laquelle l'intégrale de volume de $\mathbf{c}_i \cdot \nabla f_i$ est transformée en intégrale de surface au moyen du théorème de flux-divergence, ce qui permet d'éliminer les gradients. Ainsi, la variation temporelle d'une distribution f_i à l'intérieur d'un volume de contrôle dépend des collisions à l'intérieur du volume et des flux aux frontières. Par ailleurs, d'autres chercheurs ont noté que la méthode FVLBM originale souffrait d'instabilités et ont proposé des améliorations. Par exemple, Zarghami et al. [115] utilisent une approximation en amont du 2^e ordre basée sur la pression pour calculer les flux convectifs et l'ajout d'atténuation artificielle (artificial damping) afin améliorer la stabilité. Li et Luo [116] obtiennent aussi une stabilité accrue en proposant, entre

autres, l'utilisation de l'opérateur de collision à relaxation multiple. Quant à elle, la deuxième méthode, aussi basée sur la méthode des volumes finis, ne nécessite pas l'utilisation de modifications pour le calcul des flux ou l'ajout d'atténuation artificielle afin de maintenir la stabilité numérique. Elle diffère de la méthode FVLBM parce que ce sont les équations macroscopiques de Navier-Stokes qui sont exprimées en formulation intégrale afin d'appliquer la méthode de volumes finis. Cependant, les calculs des flux des quantités macroscopiques sont basés localement sur la LBM. C'est pourquoi cette méthode est appelée résolveur de flux par LBM (LBM Flux Solver, en anglais) [117]. En raison de sa stabilité numérique native, c'est cette méthode qui est retenue pour le développement du programme. Il faut par ailleurs noter que la méthode est utilisée par Wang et al. [118] conjointement à un réseau non-structuré autour de profils d'ailes mais le nombre de Reynolds de la simulation n'est que de 500 et une triangulation de type Delaunay est employée à la surface du profil. Pour le présent travail on désire plutôt employer des triangles plus étirés afin d'assurer un bon raffinement normal à la paroi pour bien modéliser la couche limite des écoulements turbulents.

La méthode est décrite en détail dans l'article et toutes les équations ne sont pas répétées ici. Par contre, comme elle est basée sur la correspondance du tenseur de flux de quantité de mouvement entre les équations de Navier-Stokes et la LBM, cette équivalence présentée à l'Éq. (1.44) est reproduite ci-dessous.

$$\begin{aligned} \text{Navier-Stokes : } & \frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\Pi) = 0 \\ \text{LBM : } & \Pi = \sum_{i=1}^N \mathbf{c}_i \mathbf{c}_i^T \left[f_i^{eq} + \left(1 - \frac{\Delta t}{2\Delta t_R} \right) f_i^{neq} \right] \end{aligned}$$

Selon la formulation en volumes finis, la variation temporelle de quantité de mouvement à l'intérieur d'un volume dépend du tenseur Π aux frontières de ce volume. Ce tenseur est calculé localement au moyen des distributions de la LBM. En deux dimensions, comme c'est le cas ici, la configuration d2q9 est utilisée. Or, au début d'un pas de temps donné, les seules informations connues sur le réseau sont macroscopiques, soit ρ et \mathbf{u} . (Note : le terme réseau est utilisé plutôt que maillage bien qu'en réalité les seuls réseaux LBM se trouvent aux frontières des volumes du maillage non-structuré. Le terme réseau est quand même utilisé de façon générale afin de ne pas osciller entre les deux termes.) L'interpolation de ces informations aux nœuds locaux des réseaux

d2q9 permet de reconstruire les distributions à l'équilibre f_i^{eq} . La Fig. 3.2, reproduite de la Fig. 1a de l'article, montre l'emplacement des nœuds locaux d2q9 sur le réseau non-structuré.

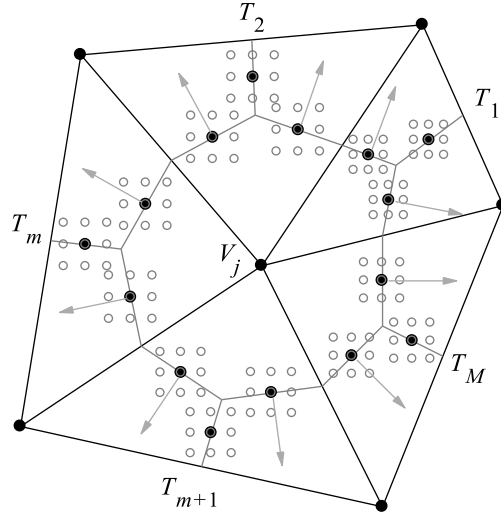


Figure 3.2 : Représentation du réseau non-structuré et des points d'interpolations.

Par contre, les distributions non-équilibrées ne peuvent pas être déterminées directement des propriétés macroscopiques. Afin de les approximer, la méthode utilise l'Éq. (1.45) découlant de l'expansion Chapman-Enskog, exprimée ici en formulation de différence finie :

$$f_i^{neq}(\mathbf{x}, t) = -\Delta t_R \frac{f_i^{eq}(\mathbf{x}, t) - f_i^{eq}(\mathbf{x} - \mathbf{c}_i \Delta t, t - \Delta t)}{\Delta t} . \quad (3.17)$$

Ainsi, les f_i^{eq} provenant de l'interpolation des propriétés macroscopiques correspondent au temps $t - \Delta t$ de l'Éq. (3.17). Ces distributions sont propagées vers le nœud central afin de calculer les propriétés macroscopiques locales au temps t par $\rho = \sum_i f_i$ et $\rho \mathbf{u} = \sum_i \mathbf{c}_i f_i$ et d'obtenir les distributions à l'équilibre correspondantes au temps t de l'Éq. (3.17). On constate qu'en réalité, les vecteurs de vitesses microscopiques des nœuds des réseaux locaux d2q9 sont les réciproques des vitesses de cette configuration, car les distributions sont propagées vers l'intérieur.

Pour cette méthode, l'application de conditions aux frontières de glissement nul sur le profil d'aile est très simple car elle se résume à imposer $u = 0$ sur les nœuds positionnés directement sur la frontière.

Il faut apporter des précisions quant au pas de temps et à la position des points d'interpolation des réseaux locaux d2q9. Tout d'abord, le pas de temps macroscopique utilisé dans la formulation en volumes finis des équations de Navier-Stokes n'est pas le même que les pas de temps locaux utilisés dans l'Éq. (3.17). Dans le premier cas, le pas de temps est le même pour tous les nœuds et est limité par une condition de type CFL, alors que dans le deuxième cas le pas de temps est égal à l'espacement local Δx du réseau d2q9 afin de produire une vitesse microscopique de $c = \Delta x / \Delta t = 1$. Le choix de cet espacement est arbitraire et Shu et al. [117] ont montré que le varier produisait les mêmes solutions pour l'écoulement dans la cavité entraînée calculé sur un réseau uniforme. Ils en ont conclu que la méthode est applicable aussi aux réseaux non-uniformes. Dans le présent travail, cet espacement est choisi afin d'être le plus grand possible tout en assurant que tous les points d'interpolations demeurent à l'intérieur de la cellule triangulaire. Ce choix est justifié par les raisons suivantes :

- Pour l'application de la méthode avec des volumes centrés sur les nœuds, comme dans le cas présent, une analyse de l'erreur pour un écoulement de Couette entre plaques parallèles a montré que l'ordre de précision spatiale était au moins de 2, c'est-à-dire que doubler le raffinement de réseau permettait de diminuer l'erreur par un facteur d'au moins 4. Or, l'analyse a aussi montré que pour un réseau donné, l'erreur augmentait si Δx était diminué. Donc il est préférable d'utiliser un Δx plus grand afin de réduire l'erreur.
- Au cours d'essais préliminaires il a été déterminé qu'attribuer au pas de temps macroscopique global la valeur du plus petit pas de temps microscopique du réseau représentait un choix adéquat pour assurer la stabilité numérique. Donc, afin d'accélérer la progression de la solution vers un état stationnaire, il est avantageux d'utiliser le plus grand pas de temps possible, ce qui implique de maximiser les Δx locaux.
- Afin d'accélérer la convergence des calculs vers un état stationnaire de l'écoulement, une approche de pas de temps local est adoptée, pour laquelle le pas de temps macroscopique de chaque nœud est déterminé par le plus petit pas de temps microscopique des triangles

formant le volume entourant le nœud. Maximiser le pas de temps microscopique permet aussi de maximiser sa contrepartie macroscopique.

Il faut noter que dans les approches de pas de temps macroscopiques local ou global, les valeurs de ces pas peuvent aussi être déterminées par l'application d'une condition CFL locale. Comme la méthode décrite plus haut assure la stabilité numérique, l'approche par CFL n'est pas utilisée.

Le modèle de turbulence retenu pour le programme est le même que pour la deuxième partie de cette thèse, afin de pouvoir comparer les résultats sur les mêmes bases. L'équation de transport du modèle S-A est encore résolue au moyen d'une approche de différence finie, cependant l'implémentation des calculs de gradients dans le programme est différente. Un schéma et les équations sont fournis dans l'article (p. 181). Par ailleurs, il faut réitérer ici le choix de la méthode de différence finie plutôt que de volume fini pour la résolution de l'équation, car la deuxième option semblerait plus naturelle étant donné que cette approche est utilisée pour le résolveur de flux par LBM. Ce choix est fait afin de limiter la mémoire d'ordinateur requise, car pour les réseaux utilisés, les nœuds peuvent avoir jusqu'à dix triangles qui composent leur volume, ce qui fait vingt points où les flux doivent être calculés. De plus, le calcul des gradients en amont est grandement simplifié par l'approche en différence finie.

3.3.2 Validation

3.3.2.1 Cavité entraînée

Le premier cas de validation choisi est celui de la cavité entraînée. Il a pour objectif de valider le programme dans le cas d'écoulements laminares et stationnaires au moyen d'un réseau dont l'aire et la proportion des cellules triangulaires ne varie pas beaucoup. Trois réseaux de raffinements différents sont utilisés, dont les nombres nœuds couvrant un côté sont 65, 129 et 257. Les réseaux de type Delaunay sont générés au moyen du logiciel Gmsh [119] et celui utilisant 65 nœuds est illustré dans l'article. Comme pour la première partie du travail, les résultats de Ghia et al. [101] sont utilisés pour les fins de comparaison. Pour des nombres de Reynolds de 100 et 1000, les résultats obtenus concordent bien, tant pour les profils de vitesse (Figs. 4-5, p. 185-186) que la localisation du tourbillon principal et des mesures de la fonction de courant et de la vorticité au centre du tourbillon (Tableau 1, p. 186).

3.3.2.2 Cylindre dans un écoulement libre

Le deuxième cas de validation consiste à simuler des écoulements stationnaires et instationnaires autour d'un cylindre. Il est choisi afin de valider le programme avec un réseau dont les cellules triangulaires varient en aire et en proportion dans le domaine. Des triangles allongés sont placés près de la paroi du cylindre spécifiquement pour produire un raffinement plus élevé dans la direction normale, comme ce sera le cas pour la validation pour un profil d'aile. Trois niveaux de réseaux sont utilisés et le moins raffiné est illustré dans l'article (Fig. 6, p. 187). Des simulations sont effectuées pour des nombres de Reynolds variant entre 20 et 200, ce qui permet de couvrir les régimes stationnaire et instationnaire. Les coefficients de portance et de traînée obtenus sont comparés avec des données d'autres auteurs produites avec des approches qui combinent la LBM et la méthode de volume fini.

Les résultats concordent généralement bien. Par contre, pour les écoulements stationnaires à $Re=20$ et $Re=40$, les coefficients de traînée obtenus avec le résolveur de flux sont plus bas (Tableau 2, p. 188). Une explication possible, fondée sur la grandeur du domaine numérique, est fournie dans l'article. Pour les écoulements instationnaires, le coefficient moyen de traînée et le nombre de Strouhal concordent très bien avec les données de référence (Tableau 3, p. 189). Le raffinement de réseau a quant à lui moins d'impact sur les résultats que pour les écoulements stationnaires, pour lesquels le nombre de Reynolds est moins élevé. Finalement, la transition entre les régimes est obtenue entre $Re=45$ et $Re=47$, ce qui concorde avec le résultat obtenu pour la première partie du travail.

3.3.2.3 Profil NACA 0012

Le troisième cas de validation concerne le thème principal de la thèse, soit la simulation d'écoulements turbulents sur profils d'ailes. Il est choisi afin de valider la combinaison d'un résolveur de flux par LBM sur réseau non-structuré avec une implémentation en différence finie du modèle de turbulence S-A. Le cas spécifique du profil NACA 0012 à $Re=5 \times 10^5$ est retenu pour comparer la méthode avec celle proposée en deuxième partie du travail.

Le domaine numérique considéré est un carré de 100m de côté, au centre duquel est placé le profil dont la corde mesure 1m. Comme l'objectif des travaux exécutés pour la rédaction de l'article ne consistait pas à étudier l'influence des caractéristiques du réseau sur les résultats mais plutôt de valider l'approche proposée de façon générale, un seul réseau est construit pour chaque angle

d'attaque simulé. Les détails des réseaux sont fournis dans l'article (p. 189). Il faut cependant souligner que les cellules triangulaires près de la paroi du profil sont plus allongées que pour le cas du cylindre afin de produire un raffinement normal élevé tout en limitant le nombre de nœuds dans la direction tangentielle, et que suite aux résultats de la deuxième partie du travail, la première rangée de nœuds est positionnée à $1/16384^{\circ}$ de corde de la paroi.

Encore une fois, la comparaison des résultats est effectuée au moyen des coefficients de forces aérodynamiques et des profils de pression et vitesse. Les coefficients aérodynamiques sont reproduits ici au Tableau 3.2. L'abréviation M-D, pour multi-domaines, désigne les résultats de la deuxième partie du travail.

Tableau 3.2 : Coefficients aérodynamiques obtenus au moyen du programme pour réseau non-structuré.

Angle [°]	C_L			C_D		
	CFL3D	M-D	Présent	CFL3D	M-D	Présent
0	0.0000	0.0000	0.0000	0.0128	0.0100	0.0095
3	0.3237	0.3257	0.3259	0.0130	0.0104	0.0099
7	0.7449	0.7457	0.7418	0.0157	0.0128	0.0122

Les coefficients de portance concordent bien avec ceux obtenus précédemment et ceux produits au moyen de CFL3D. Quant à eux, les coefficients de traînée concordent bien avec les résultats de la deuxième partie du travail mais sont encore inférieurs à ceux de CFL3D.

Les coefficients de pression pour les angles d'attaque de 0° et 7° sont similaires à ceux de CFL3D (Figs. 9-10, p. 192). De légères différences sont présentes seulement près du bord d'attaque. Pour ce qui est des profils de vitesse à 0° et 7° (Figs. 11-12, p. 193-194), les résultats obtenus sont très proches de ceux obtenus avec le programme pour multi-domaines. Similairement, la vitesse est encore un peu plus grande que celle de CFL3D dans un intervalle de distance intermédiaire à la paroi.

L'obtention de résultats qui sont très proches de ceux obtenus dans la deuxième partie du travail et qui sont aussi comparables à ceux de CFL3D indique que la méthode pour réseau non-structuré permet de simuler correctement les écoulements turbulents. De plus, cela indique que l'implémentation du modèle S-A en différence finie adaptée pour les deux types de réseaux produit des résultats similaires.

Par ailleurs, l'article soumis ne traite pas de l'écoulement pour l'angle d'attaque de 12° . Pour ce dernier, les résultats obtenus avec le programme multi-domaines ont montré des différences avec les résultats de CFL3D. Le manque de raffinement de réseau, surtout à proximité du bord d'attaque, a été identifié comme un facteur réduisant la précision des résultats. Pour cette raison, la simulation à 12° est aussi effectuée avec le présent programme afin de déterminer si les résultats peuvent être améliorés.

3.3.2.4 Résultats supplémentaires

Pour le deuxième article, la simulation de l'écoulement pour l'angle d'attaque de 12° a été effectuée avec des réseaux fournissant un espacement de $1/8192^\circ$ et $1/16384^\circ$ de corde près du profil. Malgré ce niveau de raffinement, le profil de coefficient de pression près du bord d'attaque différerait de celui produit par CFL3D. Pour cette raison, un réseau non-structuré utilisant un espacement de réseau plus petit en direction normale à la surface du profil est utilisé avec le présent programme afin de vérifier l'effet d'un raffinement plus poussé. De plus, l'avantage du raffinement directionnel de la méthode permet de limiter le nombre de nœuds le long du profil. Ainsi, le réseau construit comporte un raffinement normal de $1/32768^\circ$ de corde près du profil et des raffinements tangentiels de $1/4096^\circ$ et $1/512^\circ$ de corde aux bords d'attaque et de fuite du profil. Le réseau près du bord d'attaque est illustré à la Fig. 3.3. Le nombre total de nœuds est de 130144.

Une simulation avec le maillage raffiné est aussi effectuée en substituant la version Edwards du modèle de turbulence Spalart-Allmaras par la version originale, dans le but de vérifier l'impact du choix de modèle sur l'écoulement près du bord d'attaque du profil d'aile. La différence principale entre ces deux modèles réside dans le terme de production de viscosité turbulente, lequel est basé sur l'amplitude de la vorticit  dans le modèle original. Les coefficients aérodynamiques obtenus pour ces configurations sont fournis au Tableau 3.3 et les coefficients de pression sont fournis à la Fig. 3.4.

Tableau 3.3 : Coefficients aérodynamiques obtenus pour l'angle d'attaque de 12° au moyen du réseau raffiné à $1/32768^\circ$ de corde et des deux modèles de turbulence.

Version S-A	C_L	C_D
Edwards	1.1649	0.0234
Original	1.1498	0.0269

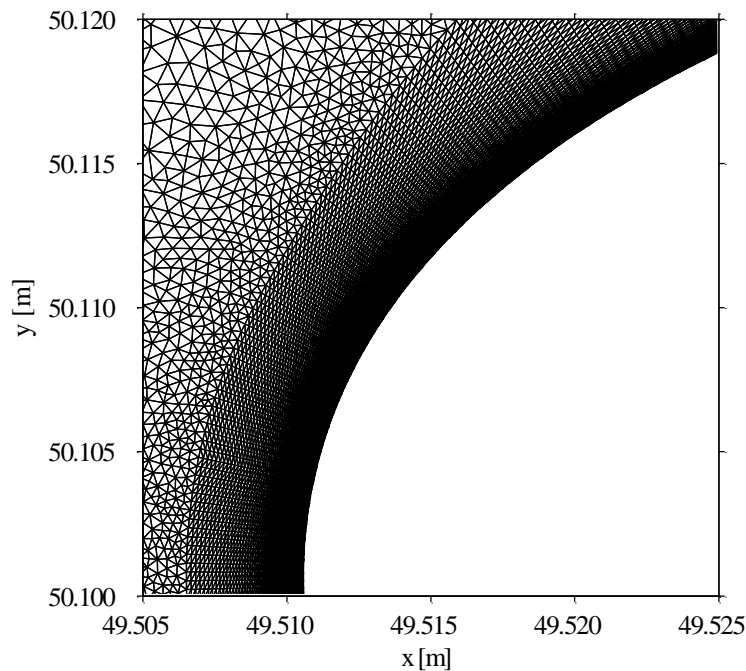


Figure 3.3 : Illustration du réseau près du bord d'attaque pour le raffinement de $1/32768^\circ$ de corde et l'angle d'attaque de 12° .

On constate tout d'abord que, comparativement au résultat obtenu avec la méthode multi-domaines du deuxième article, l'utilisation d'un réseau plus raffiné a effectivement permis de rapprocher le profil de coefficient de pression vers celui de la solution de référence, ce qui était attendu. Or, une légère ondulation, produite par la bulle de séparation marquant la transition de l'écoulement du régime laminaire vers le régime turbulent, est toujours présente. On constate aussi que cette ondulation n'est pas présente pour le résultat obtenu avec la version originale du modèle de turbulence. Ces résultats montrent donc que bien que le raffinement de réseau joue un rôle primordial sur la solution obtenue, le choix du modèle de turbulence a aussi un impact significatif. Il faut d'ailleurs en tenir compte lorsqu'on compare les coefficients aérodynamiques. Le coefficient de portance de la solution de référence du logiciel CFL3D étant de 1.1809, la version S-A Edwards du présent programme apparaît produire un meilleur résultat que la version S-A originale, or la valeur plus élevée de la portance est attribuable à la pression inférieure localisée dans la bulle de séparation et non à une meilleure concordance générale avec le profil de coefficient de pression de la solution de référence.

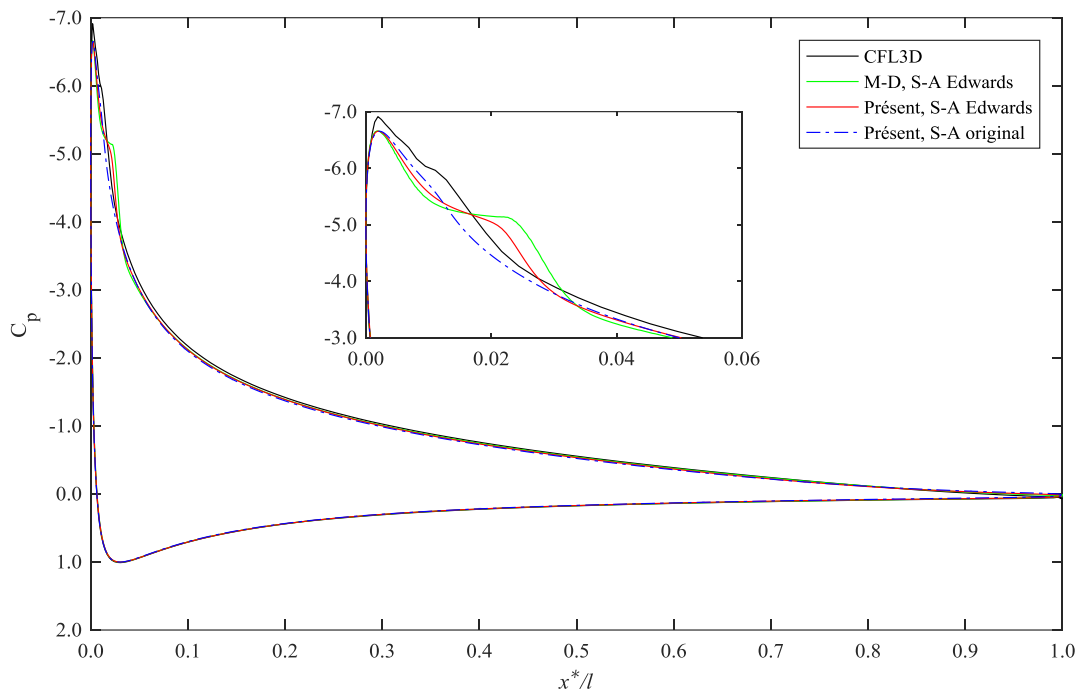


Figure 3.4 : Profils de coefficients de pression obtenus pour l'angle d'attaque de 12° au moyen du réseau raffiné à $1/32768^\circ$ de corde et des deux modèles de turbulence.

Les profils de vitesse près du bord d'attaque obtenus avec le nouveau réseau plus raffiné sont fournis à la Fig. 3.5. On constate que comparativement au résultat obtenu pour le deuxième article il y a une légère augmentation de la vitesse à une distance $y^*/l \approx 0.02$ du profil, mais qu'indépendamment du modèle de turbulence utilisé, le présent programme sous-estime encore la vitesse produite par CFL3D. Afin de vérifier si cette différence est attribuable à la géométrie du réseau, des essais ont été effectués avec un réseau utilisant un raffinement 4 fois plus fin. Les résultats obtenus n'ont pas montré de variation significative, ce qui indique que la différence observée provient de la physique modélisée par les approches numériques différentes. Dans le but de vérifier si l'écart de vitesse provient de l'effet de compressibilité de la LBM, un essai a été effectué en utilisant un nombre de Mach du réseau plus petit. Ce changement n'a pas non plus eu d'impact significatif sur les résultats produits par le présent programme. Comme le logiciel CFL3D résout les équations compressibles de Navier-Stokes et que les résultats de référence ont été obtenus avec un nombre de Mach de 0.2 [110], il est possible que la différence observée du profil de vitesse soit attribuable à l'effet de compressibilité présent dans les résultats de CFL3D plutôt que de celui présent dans les résultats de la LBM.

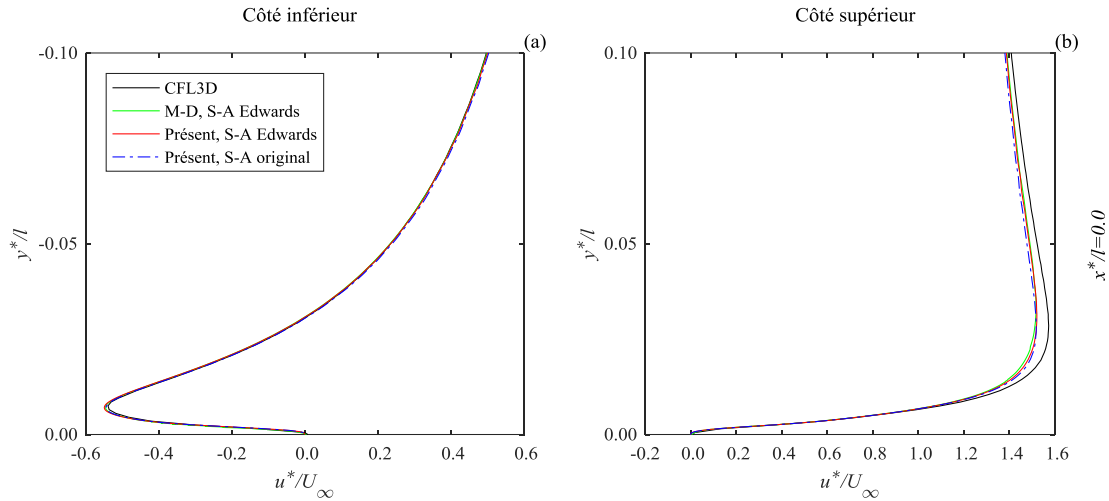


Figure 3.5 : Profils de vitesse près du bord d'attaque obtenus pour l'angle d'attaque de 12° au moyen du réseau raffiné à $1/32768^\circ$ de corde et des deux modèles de turbulence.

Puisque le raffinement au-delà de $1/32768^\circ$ de corde n'a pas produit de changement significatif des solutions obtenues, il est intéressant de discuter des valeurs de y^+ produites près du profil. Cet indicateur permet de mesurer la distance relative du réseau dans la couche limite turbulente. De façon générale, afin de bien résoudre l'écoulement dans la couche limite, le premier nœud situé près du profil doit avoir un y^+ de 1 ou moins. Cette valeur indique que le nœud se trouve dans la sous-couche visqueuse de la couche limite, qui s'étend environ jusqu'à un y^+ de 5. Pour les présentes simulations, des valeurs inférieures à 1 ont été obtenues sur la majorité du profil, sauf dans la courte partie près du bord d'attaque précédant la transition laminaire-turbulent sur le côté supérieur, où un y^+ maximum de 3 a été mesuré. Ces valeurs indiquent que le raffinement du réseau dans la direction normale au profil est adéquat pour les simulations et expliquent la faible différence des solutions obtenues avec le réseau encore plus raffiné.

Finalement, l'obtention des solutions au moyen du maillage raffiné à $1/32768^\circ$ de corde ont nécessité des temps de calcul d'environ 45 heures, pour les deux modèles de turbulence. Ce temps requis pour obtenir une convergence à la 4^e décimale des coefficients aérodynamiques est très grand et constitue une limitation du programme dans sa configuration actuelle.

3.3.3 Conclusions

Le troisième objectif de la thèse est atteint par les travaux présentés dans le troisième article. Un programme permettant de simuler des écoulements turbulents sur profil d'aile a été développé et il est basé sur un modèle LBM applicable à des réseaux non-structurés. De plus, il incorpore la résolution de l'équation d'un modèle de turbulence. Le programme est stable numériquement. Par contre, il ne permet pas de générer le réseau non-structuré et il faut recourir à un autre logiciel pour accomplir cette tâche.

Les résultats obtenus pour des simulations d'écoulements sur profil NACA 0012 confirment la validité de l'approche proposée, car la correspondance avec les résultats obtenus en deuxième partie du travail et avec ceux provenant d'un autre logiciel bien établi est satisfaisante. De plus, les simulations effectuées pour un angle d'attaque de 12° avec un réseau plus raffiné confirment une conclusion de la deuxième partie du travail, soit qu'il était nécessaire d'utiliser un réseau plus fin près du bord d'attaque afin d'améliorer les résultats, notamment les coefficients de pression.

La méthode de résolveur de flux par LBM présente quelques avantages comparativement à la méthode LBM en volume fini et aux résolveurs d'équations de Navier-Stokes. Tout d'abord, la méthode est stable numériquement sans avoir recours à des techniques de calculs en amont pour les flux. Une partie de la stabilité est bien sûr attribuable aux pas de temps très petits utilisés mais cette stabilité est aussi conférée par l'utilisation de la LBM dans le calcul des flux. En effet, la propagation des distributions le long des vecteurs de vitesses microscopiques c_i constitue une approche qui est intrinsèquement en amont. De plus, comme les points utilisés pour interpoler la vitesse servant au calcul des distributions sont distribués de tous les côtés selon la configuration d2q9, un de ces points sera toujours en amont de l'écoulement par rapport où le flux est calculé, indépendamment de la direction de la vitesse macroscopique locale. Par contre, puisque la méthode a recours à des interpolations, la présence de dissipation numérique et sa contribution à la stabilité reste à déterminer. Le peu de dissipation numérique est généralement considéré comme un avantage de la LBM standard. Deuxièmement, même si dans la méthode les volumes finis sont appliqués sur les équations macroscopiques de Navier-Stokes, la pression est déterminée sans avoir recours à une équation supplémentaire de type Poisson. En effet, puisque les tenseurs de flux de quantité de mouvement proviennent de la LBM, l'utilisation de la relation $p = \rho c_s^2$ est valide. C'est

cette relation qui a d'ailleurs été utilisée pour produire les profils de coefficients de pression présentés dans l'article.

Des limitations du programme sont aussi identifiées. Elles concernent l'usage de mémoire et la durée des calculs, effectués sur un processeur Intel i7 3.6GHz avec certaines opérations transférées sur un GPU de type GTX970 au moyen de l'outil de calcul parallèle de Matlab. Bien que les réseaux utilisés pour les simulations du troisième article soient composés d'environ 140000 nœuds, soit le quart du nombre de ceux des réseaux du programme multi-domaines, et qu'une technique de pas de temps local ait été employée, les calculs ont nécessité jusqu'à 13 heures pour atteindre une convergence à la 4^e décimale des coefficients aérodynamiques. La cause de cette limitation est la grandeur du pas de temps dans les cellules situées près du profil. En effet, le raffinement près du profil étant assuré par des nœuds situés à une hauteur de $1/16384^e$ de corde, le pas de temps minimum déterminé par les cellules triangulaires utilisant ces nœuds est de 6.36×10^{-6} . Ainsi, même si grâce à la technique de pas de temps local l'écoulement loin du profil est stabilisé plus rapidement, les petites perturbations générées près du profil prennent beaucoup de temps à voyager sur le réseau et ralentissent la convergence au niveau de précision désiré. Pour ce qui est de la limitation concernant l'usage de mémoire, elle est attribuable au choix de configuration pour la méthode de volume fini. L'utilisation de volumes centrés sur les nœuds plutôt que sur les cellules permet de simplifier le calcul des flux mais il introduit beaucoup plus d'interfaces où les flux doivent être calculés. De plus, pour chaque interface où un flux est calculé, neuf points d'interpolations sont nécessaires pour obtenir les distributions à l'équilibre de la LBM. Les coefficients d'interpolation doivent être conservés en mémoire pour tous ces points.

3.3.4 Considération supplémentaire

Une méthode alternative de construction des flux est présentée dans l'annexe de l'article. Elle est basée sur le concept de propagation des distributions à l'équilibre développé dans la première partie du travail et détaillé dans la section 3.1.1. L'utilisation de cette approche a été vérifiée et produit des résultats identiques à ceux obtenus au moyen de la méthode originale de construction des flux. Même si les deux approches diffèrent légèrement et utilisent des coefficients de relaxation différents, un développement algébrique montre qu'au final les coefficients de l'équation du tenseur de flux de quantité de mouvement Π sont les mêmes. Cette constatation démontre que le

concept de propagation des distributions à l'équilibre développé pour cette thèse n'est pas seulement applicable qu'aux conditions aux frontières.

CHAPITRE 4 DISCUSSION GÉNÉRALE

Ce chapitre fournit une perspective globale des travaux accomplis pour cette thèse. Les lacunes et les brèches de l'état actuel la LBM identifiées lors de la revue critique de la littérature, en regard du sujet d'intérêt à l'origine de la thèse, sont rappelées et les façons avec lesquelles les méthodes développées pallient ces lacunes sont expliquées. Les limitations de ces méthodes sont aussi énoncées. Tout comme pour le chapitre précédent, la discussion est organisée dans l'ordre séquentiel de réalisation des travaux.

4.1 Modèle de conditions aux frontières versatile

Un avantage de la formulation standard de la LBM repose sur la possibilité d'appliquer des conditions aux frontières sur un objet sans que le réseau utilisé ne se conforme exactement à la surface courbe de cet objet. Selon la distance existant entre les nœuds situés près de la surface et cette dernière, il est possible de déterminer les distributions f_i manquantes suite à l'opération de propagation afin d'imposer une vitesse spécifique sur la surface. De nombreuses méthodes de conditions aux frontières sont répertoriées dans la littérature, lesquelles sont basées sur une panoplie d'approches qui peuvent cependant être séparées en deux catégories, soit les méthodes qui utilisent ou non un réseau continu au travers de la surface et qui occupe aussi l'intérieur de l'objet. Pour un tel réseau, on définit la catégorie de conditions aux frontières par le terme immergé, car tel est l'état de l'objet sur le réseau. Or, aucune approche versatile, applicable aux deux types de réseaux décrits, n'est disponible dans la littérature.

La nouvelle méthode de conditions aux frontières, présentée dans le premier article publié pour cette thèse, vient colmater cette brèche. Cette méthode, basée sur la propagation de distributions à l'équilibre, permet de déterminer les distributions manquantes sur les nœuds avoisinant la surface. Sa versatilité provient de la densité utilisée pour calculer les distributions. Dans le cas immergé, elle est interpolée en utilisant les valeurs sur les nœuds situés des deux côtés de la surface, ce qui permet d'obtenir un champ de densité, ou de pression, continu au travers de cette surface et à l'intérieur de l'objet. Les distributions manquantes peuvent être obtenues par des distributions à l'équilibre des deux côtés de la surface ou on peut choisir de les calculer à l'extérieur de l'objet seulement et d'utiliser un terme de force à l'intérieur pour imposer la vitesse de l'objet solide sur

les nœuds. Dans le cas non-immérgé, la densité est approximée localement en n'utilisant que les valeurs sur les nœuds près de la surface ou en imposant une densité nominale selon le cas.

La méthode démontre par ailleurs un avantage qui constitue aussi dans certains cas une limitation. Par sa construction, elle est particulièrement applicable aux écoulements à hauts nombres de Reynolds, car sa précision est meilleure si le coefficient de relaxation $\Delta t/\Delta t_R$ s'approche de 2. Par contre, si ce coefficient est inférieur à 1, la méthode utilise de trop grandes extrapolations et une autre méthode est préférable.

4.2 Opérateur de collision stable et réseau multi-domaines

La LBM de base est formulée avec un opérateur de collision linéaire, appelé BGK. Lors du processus de collision, le rôle de cet opérateur est de diriger les distributions vers un état d'équilibre qui agit comme attracteur. Or, pour des écoulements au nombre de Reynolds élevé, le coefficient de relaxation $\Delta t/\Delta t_R$ qui détermine le chemin parcouru entre l'état initial des distributions f_i et leur attracteur f_i^{eq} s'approche de la limite de 2, ce qui représente une condition de sur-relaxation, que l'on peut aussi voir comme une extrapolation. Dans cette limite, l'opérateur BGK introduit des instabilités numériques. D'autres opérateurs, tel celui à relaxation multiple, ou MRT, permettent d'augmenter la stabilité numérique de façon significative mais éprouvent quand même des problèmes dans la limite de sur-relaxation. Or, certaines études recensées dans la littérature traitant d'écoulements à hauts nombres de Reynolds sur profils d'ailes utilisent ces opérateurs mais ne rapportent pas d'instabilités numériques, ce qui force à se demander si des stratégies additionnelles de stabilisation sont employées ou si la dissipation numérique des modèles LBM utilisés stabilise les calculs. Dans le cas de la LBM appliquée à un réseau multi-domaines, la stabilité de ces opérateurs reste à vérifier.

Le programme développé pour la publication du deuxième article répond à cette lacune. Il est construit pour simuler des écoulements à hauts nombres de Reynolds en utilisant la configuration de réseau multi-domaine. Comme la méthode LBM standard est utilisée sur chaque réseau, la dissipation numérique ne constitue pas un facteur de stabilisation considérable. Or, des simulations préliminaires avec les opérateurs BGK, MRT et même l'opérateur régularisé produisent des solutions instables. Pour cette raison, l'opérateur de collision en cascade est incorporé au programme. Sa construction, appuyée sur le concept des moments centraux, lui confère une grande

stabilité. Cette dernière est vérifiée par les simulations exécutées pour le deuxième article, qui ne présentent aucune instabilité. L'utilisation conjointe d'un réseau multi-domaines et de l'opérateur en cascade constitue donc une approche adéquate pour simuler des écoulements à hauts nombres de Reynolds.

Par contre, la méthode de transfert d'information intégrée au programme a été développée pour l'opérateur de collision BGK. Bien qu'aucun problème n'en découle dans les simulations et que les champs macroscopiques soient continus au travers des interfaces, une validation théorique plus formelle devrait être exécutée, en comparant avec une approche conçue spécifiquement pour l'opérateur en cascade.

4.3 Modèle de turbulence Spalart-Allmaras et réseau multi-domaines

Le thème principal de cette thèse est la simulation d'écoulements turbulents sur profils d'ailes. Or, dans sa forme native la LBM ne contient aucune disposition pour traiter explicitement de la turbulence. Dans le cas d'écoulements bidimensionnels, comme ceux considérés pour ce travail, toutes les échelles de turbulence doivent être modélisées. Pour ce faire, il existe des approches basées sur des équations différentielles modélisant le transport de quantités associées à la turbulence. Par exemple, le modèle Spalart-Allmaras est fondé sur le transport d'une viscosité turbulente. Des études recensées dans la littérature, une seule utilise ce modèle conjointement à la LBM appliquée à un réseau multi-domaines et elle est considérée comme incomplète, tant pour la qualité des résultats que pour la description de l'implémentation du modèle S-A.

Le programme développé pour le deuxième article pallie cette lacune. Une formulation en différence finie de l'équation du modèle S-A y est décrite en détail, tout comme la façon dont le transfert d'information aux interfaces des réseaux est exécuté. La façon d'utiliser la viscosité turbulente dans l'opérateur de collision de la LBM afin d'y inclure les effets de la turbulence est aussi détaillée. L'article contient suffisamment d'information pour que la méthode proposée soit reproduite. Des simulations exécutées pour les profils d'ailes NACA 0012, S1223 et E387 permettent de valider la méthode en produisant des résultats qui concordent bien avec les références choisies. Tant les coefficients aérodynamiques, que les profils de pression et de vitesse sont en accord avec ceux obtenus avec un résolveur établi des équations Navier-Stokes.

Cependant, la méthode, telle que présentée dans l'article, comporte deux limitations. Premièrement, la nécessité d'utiliser un réseau plus fin est identifiée afin d'améliorer les résultats lorsque les écoulements considérés comportent des sections détachées ou s'approchent des conditions de décrochage aérodynamique. Or, le programme ne permet pas un raffinement localisé sur le profil car le même espacement de réseau doit être utilisé sur tout le périmètre. Le traitement du transfert des distributions aux interfaces de réseaux situées sur la surface d'un objet doit être le sujet de recherches plus approfondies. Deuxièmement, les temps de calculs requis pour obtenir des solutions stationnaires sont trop grands. Conséquemment, des améliorations doivent être apportées au programme, tant pour rendre l'exécution des calculs plus rapide que pour y inclure des techniques d'accélération de convergence.

4.4 Écoulements turbulents, LBM et réseaux non-structurés

Le thème principal de cette thèse comporte un critère spécifique quant aux géométries des objets étant assujettis aux écoulements turbulents et à l'adaptabilité des réseaux utilisés pour simuler ces écoulements. En effet, il est établi que les réseaux doivent pouvoir fournir un raffinement local même si l'objet ne présente pas une géométrie simple. Ce critère émane de l'idée instigatrice de cette thèse qui consistait à simuler la formation de glace sur des profils d'ailes et à calculer les écoulements sur des profils givrés. C'est pour cette raison que le réseau multi-domaine constitue un élément du deuxième article. Or, en regard de la limitation identifiée dans l'implémentation de ce type de réseau dans le programme développé pour la présente thèse, c'est-à-dire que le raffinement local sur la surface du profil est difficile, le réseau de type non-structuré est considéré. Cependant, il n'y a pas d'études dans la littérature qui traite de l'utilisation d'un modèle de turbulence avec une méthode LBM pour réseau non-structuré.

Le programme développé pour le troisième article comble ce vide. Dans la littérature, deux approches existent pour appliquer la LBM à un réseau non-structuré, soit la méthode LBM par volumes finis et la méthode de résolveur de flux par LBM. Étant donné les conclusions tirées par différents auteurs sur la stabilité numérique de ces méthodes, la seconde est retenue et intégrée au programme. De plus, le modèle de turbulence S-A, utilisé pour produire les résultats du deuxième article, est aussi intégré au programme. Il y est encore formulé par une méthode de différence finie, dont les détails sont fournis dans l'article. Les détails complets de la méthode de résolveur de flux par LBM y sont aussi fournis afin que la méthode globale puisse être reproduite. Le seul élément

absent de l'article est l'algorithme de génération du réseau non-structuré, car un logiciel tiers est utilisé pour accomplir cette tâche. Des simulations exécutées pour un profil NACA 0012 pour les mêmes conditions que dans le deuxième article permettent de valider la méthode. En effet, les résultats produits par les deux méthodes proposées dans cette thèse concordent bien ensemble et avec les résultats numériques provenant de la littérature. De plus, le manque de précision des résultats présentés dans le deuxième article pour le profil NACA 0012 à un angle d'attaque de 12° est réduit par la méthode proposée pour réseaux non-structurés. En effet, la méthode permet d'utiliser un réseau plus raffiné près du bord d'attaque et ainsi d'obtenir une meilleure concordance du profil de coefficient de pression.

Par contre, il faut aussi discuter des limitations du programme. Tout d'abord, le temps de calcul constitue aussi un problème, et ce même si une technique d'accélération de convergence est utilisée. Ce problème provient de la grandeur des pas de temps utilisés, lesquels sont ajustés en fonction des plus petites cellules du réseau situées près de la surface du profil. Les petites perturbations de l'écoulement générées près de la surface prennent beaucoup d'itérations, donc de temps, à faire sentir leur effet sur le reste du réseau. En deuxième lieu, la construction du modèle au moyen de volumes de contrôles centrés sur les nœuds du réseau implique qu'un plus grand nombre de surfaces utilisées pour les calculs des flux sont présentes que si les volumes de contrôle étaient centrés sur les cellules. La mémoire utilisée pour stocker les coefficients d'interpolation et calculer les flux est donc très grande. L'approche choisie pour le programme présente un avantage de simplicité d'implémentation aux dépens d'un besoin de mémoire augmenté.

CHAPITRE 5 CONCLUSION ET RECOMMANDATIONS

La dynamique numérique des fluides constitue un domaine de l'ingénierie qui revêt une très grande importance dans la conception de divers systèmes, dont les automobiles et les avions. En effet, elle permet de calculer les caractéristiques aérodynamiques telles la traînée et la portance, lesquelles ont un impact majeur sur la performance des véhicules. Même si les simulations par ordinateur ne peuvent pas remplacer complètement les expériences en soufflerie, elles procurent l'avantage d'accélérer la phase de conception en validant les concepts préliminaires. Elles permettent aussi d'évaluer la dégradation des performances aérodynamiques dans le cas où un véhicule subit une modification ou une détérioration, comme la formation de glace sur une aile d'avion.

Même s'il existe des logiciels commerciaux établis de dynamique numérique des fluides, la recherche dans ce domaine est très active car il y a toujours place à des améliorations. C'est dans ce contexte que cette thèse s'inscrit. Au cours des vingt-cinq dernières années, une méthode alternative aux approches de résolution des équations de Navier-Stokes a émergé, soit la méthode de Boltzmann sur réseau. Cette méthode se différencie des autres parce qu'elle ne traite pas des écoulements de fluides directement du point de vue des milieux continus mais utilise plutôt l'échelle mésoscopique pour observer les fluides, c'est-à-dire que son point de vue est probabiliste. En effet, elle est basée sur la collision et la propagation de particules qui sont représentées par des distributions de probabilité. Cette méthode trouve son origine dans la théorie cinétique des gaz et l'équation de Boltzmann. Comparativement aux autres méthodes, elle présente les avantages d'être explicite, de ne pas recourir à la résolution d'une équation différentielle non-linéaire, de ne pas nécessiter une seconde équation pour le champ de pression et, selon un constat qui semble généralisé, être plus facilement parallélisable.

Cette thèse avait pour objectif de poursuivre le développement continu de la méthode de Boltzmann sur réseau en l'utilisant pour simuler des écoulements turbulents sur profils d'ailes d'avions. À l'origine, le point de départ du projet consistait à développer un programme de simulation de formation de glace sur les profils d'ailes d'avions en utilisant la méthode de Boltzmann sur réseau à la fois pour calculer l'écoulement aérodynamique et pour modéliser le transport des gouttelettes d'eau qui produisent la glace au contact de l'aile. Or, une évaluation de l'état de la recherche a indiqué que le premier thème nécessitait suffisamment d'attention pour constituer le sujet principal de la thèse. Par contre, le phénomène de formation de glace n'a pas été évacué complètement,

puisque les travaux ont été conduits de façon à ce que les méthodes et programmes développés pour cette thèse puissent être utilisés éventuellement pour simuler des écoulements aérodynamiques sur des profils d'ailes contaminés par la présence de glace.

Trois contributions principales ont découlé des travaux. Premièrement, la première phase de la recherche consistait à développer les connaissances de la méthode et à développer l'habileté à créer des programmes de simulations. Dans ce contexte, une nouvelle méthode de conditions aux frontières a été proposée et validée, ce qui a constitué le sujet du premier article publié. Deuxièmement, en regard des lacunes observées dans l'état de la recherche, un programme a été conçu afin de simuler des écoulements turbulents en alliant trois ingrédients principaux, soit la méthode de Boltzmann sur un réseau en configuration multi-domaines, l'opérateur de collision en cascade et le modèle de turbulence Spalart-Allmaras. Cette configuration n'avait pas été proposée antérieurement et a constitué le sujet du deuxième article publié. Les validations ont montré que les résultats produits en termes de caractéristiques aérodynamiques des profils d'ailes étudiés étaient satisfaisants, dans la mesure où le raffinement du réseau utilisé était adéquat. Troisièmement, suite à ce dernier constat, une approche alternative au réseau multi-domaine a été abordée. En effet, la possibilité de simuler des écoulements turbulents au moyen de la méthode de Boltzmann sur réseau non-structuré a été explorée. Un programme de simulation en a découlé, proposant une approche jusque-là absente de la littérature, soit l'utilisation de la méthode récente de résolveur de flux par LBM, conjointement à un réseau non-structuré fournissant un raffinement directionnel près du profil et au modèle de turbulence Spalart-Allmaras. Les détails de cette approche ainsi que les résultats produits ont été présentés dans le troisième article. Ces résultats se sont d'ailleurs montrés encore une fois satisfaisants. De plus, des simulations complémentaires ont été exécutées et ont montré que le raffinement directionnel du réseau représentait un avantage comparativement au raffinement uniforme de la méthode du deuxième article, en permettant d'améliorer les résultats sans augmenter le nombre de nœuds du réseau.

Finalement, on peut affirmer que les méthodes développées dans cette thèse et les résultats obtenus permettent de répondre de façon affirmative aux questions de recherche, soit que la LBM est appropriée pour simuler des écoulements turbulents sur profils d'ailes et que la simulation d'écoulements sur profils givrés est envisageable.

5.1 Recommandations

Cette thèse se termine par des recommandations découlant d'observations et constatations faites lors de la réalisation des travaux, lesquelles pourraient être utiles pour des recherches futures.

1. Il est mentionné plus haut qu'un avantage de la LBM consiste à ne pas devoir résoudre une équation différentielle non-linéaire. Or, l'implémentation du modèle de turbulence Spalart-Allmaras est basée sur une telle approche et déroge donc de la simplicité inhérente de la LBM. Comme il existe des modèles LBM développés pour simuler le phénomène d'advection-diffusion [120], il serait intéressant d'étudier la possibilité d'utiliser une telle approche pour concevoir un modèle de turbulence entièrement basé sur la collision et propagation de distributions. Une idée similaire avait d'ailleurs été proposée par Succi et al. [121] pour créer un pendant LBM du modèle de turbulence $k-\varepsilon$. Cependant, aucune suite ne semble avoir été donnée à cette idée dans la littérature.
2. Une limitation des programmes conçus pour les deuxième et troisième parties du travail découle du très grand temps de calcul requis pour obtenir des solutions stationnaires produisant une convergence à la quatrième décimale des coefficients aérodynamiques. Hormis l'amélioration de la construction des programmes et le passage éventuel vers un autre langage de programmation que Matlab, l'implémentation de fonctions de paroi pourrait permettre de limiter le raffinement nécessaire près des profils d'ailes. Le temps de calcul serait réduit parce qu'il y aurait moins de nœuds sur le réseau et que le pas de temps utilisé serait plus grand.
3. Le programme de la deuxième partie du travail utilise un réseau multi-domaines pour lequel le réseau le plus raffiné couvre tout le périmètre du profil d'aile. Cette configuration est utilisée parce que des essais effectués avec des raffinements différents près de la paroi ont produit des discontinuités des champs macroscopiques aux interfaces. Comme la méthode choisie pour le transfert d'information en est une d'interpolation, il serait intéressant d'étudier le problème plus en détail et de voir si des améliorations pourraient être apportées à l'algorithme d'interpolation.
4. L'opérateur de collision en cascade est utilisé dans le programme multi-domaines parce qu'il est très stable. Or, Geier et al. [33] affirment aussi que cet opérateur «ne vise pas à

éliminer les hautes fréquences de l'écoulement mais de les traiter de façon précise sans affecter la précision des basses fréquences», et que cette méthode est correcte pour «accomplir une séparation des échelles résolues et non-résolues dans la simulation d'écoulements turbulents». Geier et al. affirment aussi que son objectif est de «capturer correctement le comportement physique des ondes les plus courtes résolues par le raffinement du réseau». Ces affirmations font en sorte que cette méthode s'apparente à la méthode de simulation des grandes échelles (LES) pour les écoulements turbulents. Il serait donc très intéressant, si les ressources computationnelles le permettent, de concevoir un programme de simulation 3D pour écoulements turbulents sur ailes d'avion comparant l'utilisation stricte de l'opérateur de collision en cascade avec un autre opérateur de collision couplé à un modèle LES, tel le modèle de Smagorinsky. Ce type de simulation a déjà été effectué pour l'écoulement autour d'une sphère [58] et il serait intéressant d'étendre la méthode à une aile.

RÉFÉRENCES

- [1] Chapman S, Cowling TG. The mathematical theory of non-uniform gases. Third. Cambridge University Press; 1970.
- [2] The Editors of Encyclopædia Britannica. Kinetic theory of gases. Encycl Br 2016. <https://www.britannica.com/science/kinetic-theory-of-gases> (accessed November 11, 2016).
- [3] Mason EA. Gas. Encycl Br 2016. <https://www.britannica.com/science/gas-state-of-matter/Kinetic-theory-of-gases#ref507018> (accessed November 11, 2016).
- [4] The Editors of Encyclopædia Britannica. Perfect gas. Encycl Br 2016. <https://www.britannica.com/science/perfect-gas> (accessed November 11, 2016).
- [5] Müller-Kirsten HJW. Statistical Mechanics of an Ideal Gas (Maxwell). Basics Stat. Phys. Second, World Scientific; 2013, p. 11–25. doi:10.1142/9789814449540_0002.
- [6] Viggen EM. The lattice Boltzmann method: fundamentals and acoustics. Norwegian University of Science and Technology, 2014.
- [7] Succi S. The lattice Boltzmann equation for fluid dynamics and beyond. Oxford University Press; 2001.
- [8] Bhatnagar PL, Gross EP, Krook M. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. Phys Rev 1954;94:511–25. doi:<https://doi.org/10.1103/PhysRev.94.511>.
- [9] Cédric Villani. A review of Mathematical Topics in Collisional Kinetic Theory. In: Friedlander S, Serre D, editors. Handb. Math. Fluid Dyn. Vol. 1, Elsevier Science; 2002, p. 71–305.
- [10] Wolf-Gladrow DA. Lattice-Gas Cellular Automata and Lattice Boltzmann Models - An Introduction. Berlin: Springer; 2005.
- [11] Gardner M. Mathematical Games. The fantastic combinations of John Conway’s new solitaire game “life”. Sci Am 1970;223:120–3.

- [12] Wolfram S. A new kind of science. Wolfram Media Inc; 2002.
- [13] Hardy J, Pomeau Y, De Pazzis O. Time evolution of a two-dimensional classical lattice system. *Phys Rev Lett* 1973;31:276–9. doi:10.1103/PhysRevLett.31.276.
- [14] Hardy J, De Pazzis O, Pomeau Y. Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions. *Phys Rev A* 1976;13:1949–61. doi:10.1103/PhysRevA.13.1949.
- [15] Viggen EM. The lattice Boltzmann method with applications in acoustics. Norwegian University of Science and Technology, 2009.
- [16] Frisch U, Hasslacher B, Pomeau Y. Lattice as Automata for the Navier-Stokes Equation. *Phys Rev Lett* 1986;56:1505–8. doi:10.1103/PhysRevLett.56.1505.
- [17] Chopard B, Dupuis A, Masselot A, Luthi P. Cellular automata and lattice Boltzmann techniques: an approach to model and simulate complex systems. *Adv Complex Syst* 2002;5:103–246. doi:10.1142/S0219525902000602.
- [18] D’Humières D, Lallemand P. Numerical simulations of hydrodynamics with lattice gas automata in two dimensions. *Complex Syst* 1987;1:599–632.
- [19] McNamara GR, Zanetti G. Use of the Boltzmann equation to simulate lattice-gas automata. *Phys Rev Lett* 1988;61:2332–5. doi:10.1103/PhysRevLett.61.2332.
- [20] Higuera FJ, Jiménez J. Boltzmann approach to lattice gas simulations. *EPL (Europhysics Lett)* 1989;9:663–8.
- [21] Koelman JMVA. A Simple Lattice Boltzmann Scheme for Navier-Stokes Fluid Flow. *Europhys Lett* 1991;15:603–7. doi:10.1209/0295-5075/15/6/007.
- [22] Qian YH, D’Humières D, Lallemand P. Lattice BGK models for Navier-Stokes equation. *Europhys Lett* 1992;17:479–84.
- [23] He X, Luo L-S. Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation. *Phys Rev E* 1997;56:6811–7. doi:10.1103/PhysRevE.56.6811.
- [24] Lätt J. Hydrodynamic limit of lattice Boltzmann equations. Université de Genève, 2007.

- [25] Chen S, Doolen GD. Lattice Boltzmann method for fluid flows. *Annu Rev Fluid Mech* 1998;30:329–64. doi:10.1146/annurev.fluid.30.1.329.
- [26] Zou Q, He X. On pressure and velocity flow boundary conditions and bounceback for the lattice Boltzmann BGK model. *Phys Fluids* 1997;9:1–19.
- [27] Bouzidi M, Firdaouss M, Lallemand P. Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Phys Fluids* 2001;13:3452–9. doi:10.1063/1.1399290.
- [28] Guo Z, Zheng C, Shi B. An extrapolation method for boundary conditions in lattice Boltzmann method. *Phys Fluids* 2002;14:2007. doi:10.1063/1.1471914.
- [29] Feng Z-G, Michaelides EE. The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems. *J Comput Phys* 2004;195:602–28. doi:10.1016/j.jcp.2003.10.013.
- [30] D’Humières D. Generalized lattice-Boltzmann equations. *Rarefied gas Dyn. theory simulations*, American Institute of Aeronautics and Astronautics, Inc.; 1994, p. 450–8.
- [31] Ansumali S, Karlin I V. Entropy function approach to the lattice Boltzmann method. *J Stat Phys* 2002;107:291–308.
- [32] Geier MC. Ab initio derivation of the cascaded lattice boltzmann automaton. University of Freiburg, 2006.
- [33] Geier M, Greiner A, Korvink JG. Cascaded digital lattice Boltzmann automata for high Reynolds number flow. *Phys Rev E* 2006;73:66705.
- [34] Lagrava D, Malaspinas O, Latt J, Chopard B. Advances in multi-domain lattice Boltzmann grid refinement. *J Comput Phys* 2012;231:4808–22. doi:10.1016/j.jcp.2012.03.015.
- [35] Imamura T, Suzuki K, Nakamura T, Yoshida M. Flow simulation around an airfoil by lattice Boltzmann method on generalized coordinates. *AIAA J* 2005;43:1968–73. doi:10.2514/1.7554.
- [36] He X, Doolen G. Lattice Boltzmann method on curvilinear coordinates system: flow around a circular cylinder. *J Comput Phys* 1997;134:306–15. doi:10.1006/jcph.1997.5709.
- [37] Stiebler M, Tölke J, Krafczyk M. An upwind discretization scheme for the finite volume lattice Boltzmann method. *Comput Fluids* 2006;35:814–9.

doi:10.1016/j.compfluid.2005.09.002.

- [38] Moreau S, Sanjos M, Kim M, P F. Direct simulation of trailing-edge noise generated by a controlled diffusion airfoil using a lattice-Boltzmann method. Seventh Int. Symp. Turbul. Shear Flow Phenom., Ottawa: Turbulence and Shear Flow Phenomena; 2011, p. 1–6.
- [39] Sanjosé M, Moreau S. Direct numerical simulation of self-noise of an installed control-diffusion airfoil. *Can Acoust* 2011;39:2–3.
- [40] Holman DM, Brionnaud RM, Modena MC. Aerodynamic analysis of the 2nd high lift prediction workshop by a lattice-Boltzmann method solver. 32nd AIAA Appl. Aerodyn. Conf., Atlanta, GA: AIAA; 2014. doi:10.2514/6.2014-2568.
- [41] König B, Fares E, Nölting S. PowerFLOW Analysis HiLiftPW-2 Configuration. 52nd AIAA Aerosp Sci Meet 2014.
- [42] Satti R, Li Y, Shock R, Duncan B. Computational analysis of a wingtip vortex in the near-field using LBM-VLES approach. 49th AIAA Aerosp. Sci. Meet. Incl. New Horizons Forum Aerosp. Expo., Orlando, FL: AIAA; 2011, p. 1–12.
- [43] Satti R, Li Y, Shock R, Noelting S. Unsteady flow analysis of a multi-element airfoil using lattice Boltzmann method. *AIAA J* 2012;50. doi:10.2514/1.J050906.
- [44] Koenig B, Fares E, Jammalamadaka A, Li Y. Investigation of the NACA 4412 trailing Edge separation using a lattice-Boltzmann approach. 44th AIAA Fluid Dyn. Conf., American Institute of Aeronautics and Astronautics; 2014. doi:doi:10.2514/6.2014-3324.
- [45] König B, Fares E, Broeren A. Lattice-Boltzmann analysis of three-dimensional ice shapes on a NACA 23012 airfoil. *SAE Tech Pap* 2015. doi:10.4271/2015-01-2084.
- [46] Li Y, Jammalamadaka A, Gopalakrishnan P, Zhang R, Chen H. Computation of high-subsonic and transonic flows by a lattice Boltzmann method. 54th AIAA Aerosp. Sci. Meet., San Diego, CA: American Institute of Aeronautics and Astronautics, AIAA SciTech Forum; 2016, p. 1–15. doi:10.2514/6.2016-0043.
- [47] Filippova O, Succi S, Mazzocco F, Arrighetti C, Bella G, Hänel D, et al. Multiscale lattice Boltzmann schemes with turbulence modeling. *J Comput Phys* 2001;170:812–29. doi:10.1006/jcph.2001.6764.

- [48] Succi S, Filippova O, Smith G, Kaxiras E. Applying the lattice Boltzmann equation to multiscale fluid problems. *Comput Sci ...* 2001;3:26–37.
- [49] Teixeira CM. Incorporating turbulence models into the lattice-Boltzmann method. *Int J Mod Phys C* 1998;9:1159–75. doi:10.1142/S0129183198001060.
- [50] Premnath KN, Pattison MJ, Banerjee S. Dynamic subgrid scale modeling of turbulent flows using lattice-Boltzmann method. *Phys A Stat Mech Its Appl* 2009;388:2640–58. doi:http://dx.doi.org/10.1016/j.physa.2009.02.041.
- [51] Li Y, Shock R, Zhang R, Chen H, Shih T. Simulation of flow over an iced airfoil by using a lattice-Boltzmann method. 43rd AIAA Aerosp. Sci. Meet. Exhib., Reno, NV: American Institute of Aeronautics and Astronautics; 2005, p. 1–8. doi:http://dx.doi.org/10.2514/6.2005-1103.
- [52] Li Y, Shock R, Zhang R, Chen H. Numerical study of flow past an impulsively started cylinder by the lattice-Boltzmann method. *J Fluid Mech* 2004;519:273–300. doi:10.1017/S0022112004001272.
- [53] Chen H, Filippova O, Hoch J, Molvig K, Shock R, Teixeira C, et al. Grid refinement in lattice Boltzmann methods based on volumetric formulation. *Phys A Stat Mech Its Appl* 2006;362:158–67. doi:10.1016/j.physa.2005.09.036.
- [54] Chi X, Li Y, Chen H, Addy H, Choo Y, Shih T. A comparative study using CFD to predict iced airfoil aerodynamics. *AIAA Pap* 2005;1371:1–12. doi:10.2514/6.2005-1371.
- [55] Chen X-P. Applications of lattice Boltzmann method to turbulent flow around two-dimensional airfoil. *Eng Appl Comput Fluid Mech* 2012;6:572–80.
- [56] Li K, Zhong C, Zhuo C, Cao J. Non-body-fitted Cartesian-mesh simulation of highly turbulent flows using multi-relaxation-time lattice Boltzmann method. *Comput Math with Appl* 2012;63:1481–96. doi:10.1016/j.camwa.2012.03.080.
- [57] Si H, Shi Y. Study on lattice Boltzmann method/large eddy simulation and its application at high Reynolds number flow. *Adv Mech Eng* 2015;7:1–8. doi:10.1177/1687814015573829.
- [58] Ishida T. Lattice Boltzmann method for aeroacoustic Simulations with block-structured Cartesian grid. 54th AIAA Aerosp. Sci. Meet., American Institute of Aeronautics and

- Astronautics; 2016, p. 1–8. doi:doi:10.2514/6.2016-0259.
- [59] Imamura T, Suzuki K, Nakamura T, Yoshida M. Acceleration of steady-state lattice Boltzmann simulations on non-uniform mesh using local time step method. *J Comput Phys* 2005;202:645–63. doi:10.1016/j.jcp.2004.08.001.
 - [60] Filippova O, Hänel D. Lattice-Boltzmann simulation of gas-particle flow in filters. *Comput Fluids* 1997;26:697–712.
 - [61] Mei R, Luo L-S, Shyy W. An accurate curved boundary treatment in the lattice Boltzmann method. *J Comput Phys* 1999;155:307–30. doi:10.1006/jcph.1999.6334.
 - [62] Lallemand P, Luo L-S. Lattice Boltzmann method for moving boundaries. *J Comput Phys* 2003;184:406–21. doi:10.1016/S0021-9991(02)00022-0.
 - [63] Rohde M, Kandhai D, Derksen J, Van den Akker H. Improved bounce-back methods for no-slip walls in lattice-Boltzmann schemes: Theory and simulations. *Phys Rev E* 2003;67:66703. doi:10.1103/PhysRevE.67.066703.
 - [64] Ginzburg I, d’Humières D. Multireflection boundary conditions for lattice Boltzmann models. *Phys Rev E* 2003;68:66614.
 - [65] Chun B, Ladd a. Interpolated boundary condition for lattice Boltzmann simulations of flows in narrow gaps. *Phys Rev E* 2007;75:66705. doi:10.1103/PhysRevE.75.066705.
 - [66] Verberg R, Ladd a. Lattice-Boltzmann Model with Sub-Grid-Scale Boundary Conditions. *Phys Rev Lett* 2000;84:2148–51. doi:10.1103/PhysRevLett.84.2148.
 - [67] Rohde M, Derksen JJ, Van Den Akker HEA. Volumetric method for calculating the flow around moving objects in lattice-Boltzmann schemes. *Phys Rev E - Stat Nonlinear, Soft Matter Phys* 2002;65:1–11. doi:10.1103/PhysRevE.65.056701.
 - [68] Kao P-H, Yang R-J. An investigation into curved and moving boundary treatments in the lattice Boltzmann method. *J Comput Phys* 2008;227:5671–90. doi:10.1016/j.jcp.2008.02.002.
 - [69] Filippova O, Hänel D. Grid refinement for lattice-BGK models. *J Comput Phys* 1998;147:219–28. doi:10.1006/jcph.1998.6089.
 - [70] Kuo L-S, Chen P-H. A unified approach for nonslip and slip boundary conditions in the

- lattice Boltzmann method. *Comput Fluids* 2009;38:883–7. doi:10.1016/j.compfluid.2008.09.008.
- [71] Kang J, Kang S, Suh YK. A dynamic boundary model for implementation of boundary conditions in lattice-Boltzmann method. *J Mech Sci Technol* 2008;22:1192–201. doi:10.1007/s12206-008-0307-y.
- [72] Tiwari A, Vanka SP. A ghost fluid Lattice Boltzmann method for complex geometries. *Int J Numer Methods Fluids* 2012;69:481–98. doi:10.1002/fld.2573.
- [73] Chen L, Yu Y, Hou G. Sharp-interface immersed boundary lattice Boltzmann method with reduced spurious-pressure oscillations for moving boundaries. *Phys Rev E* 2013;87:53306. doi:10.1103/PhysRevE.87.053306.
- [74] Chen S, Bao S, Liu Z, Li J, Yi C, Zheng C. A heuristic curved-boundary treatment in lattice Boltzmann method. *Europhys Lett* 2010;92:54003. doi:10.1209/0295-5075/92/54003.
- [75] Chen H, Teixeira C, Molvig K. Realization of fluid boundary conditions via discrete Boltzmann dynamics. *Int J Mod Phys C* 1998;9:1281–92. doi:10.1142/S0129183198001151.
- [76] Rohde M. Extending the lattice-Boltzmann method: Novel techniques for local grid refinement and boundary conditions. Delft University of Technology, 2004.
- [77] Li Y. An improved volumetric LBM boundary approach and its extension for sliding mesh simulation. Iowa State University, 2011.
- [78] Li Y, Zhang R, Shock R, Chen H. Prediction of vortex shedding from a circular cylinder using a volumetric Lattice-Boltzmann boundary approach. *Eur Phys J Spec Top* 2009;171:91–7. doi:10.1140/epjst/e2009-01015-9.
- [79] Yu H, Chen X, Wang Z, Deep D, Lima E, Zhao Y, et al. Mass-conserved volumetric lattice Boltzmann method for complex flows with willfully moving boundaries. *Phys Rev E* 2014;89:63304. doi:10.1103/PhysRevE.89.063304.
- [80] Zhang J, Johnson PC, Popel AS. An immersed boundary lattice Boltzmann approach to simulate deformable liquid capsules and its application to microscopic blood flows. *Phys Biol* 2007;4:285–95. doi:10.1088/1478-3975/4/4/005.

- [81] Strack OE, Cook BK. Three-dimensional immersed boundary conditions for moving solids in the lattice-Boltzmann method. *Int J Numer Methods Fluids* 2007;55:103–25. doi:10.1002/fld.
- [82] Shu C, Liu N, Chew Y, Lu Z. Numerical simulation of fish motion by using lattice Boltzmann-immersed boundary velocity correction method. *J Mech Sci Technol* 2007;21:1139–49.
- [83] Shu C, Wu J. A new immersed boundary-lattice Boltzmann method and its application to incompressible flows. *Mod Phys Lett B* 2009;23:261–4. doi:10.1142/S0217984909018151.
- [84] Wu J, Shu C, Zhang YH. Simulation of incompressible viscous flows around moving objects by a variant of immersed boundary-lattice Boltzmann method. *Int J Numer Methods Fluids* 2010;62:327–54. doi:10.1002/fld.
- [85] Tian F-B, Luo H, Zhu L, Liao JC, Lu X-Y. An efficient immersed boundary-lattice Boltzmann method for the hydrodynamic interaction of elastic filaments. *J Comput Phys* 2011;230:7266–83. doi:10.1016/j.jcp.2011.05.028.
- [86] Rojas R, Seta T, Hayashi K, Tomiyama A. Immersed boundary-finite difference lattice Boltzmann method using two relaxation times. In: CSIRO, editor. Ninth Int. Conf. CFD Miner. Process Ind., 2012, p. 1–6.
- [87] Seta T, Rojas R, Hayashi K, Tomiyama A. Implicit-correction-based immersed boundary–lattice Boltzmann method with two relaxation times. *Phys Rev E* 2014;89:23307. doi:10.1103/PhysRevE.89.023307.
- [88] Nishida H, Meichin Y. Seamless immersed boundary lattice Boltzmann method for incompressible flow simulation. Seventh Int. Conf. Comput. Fluid Dyn., Hawaii: ICCFD; 2012, p. 1–19.
- [89] Peng Y, Shu C, Chew YT, Niu XD, Lu XY. Application of multi-block approach in the immersed boundary–lattice Boltzmann method for viscous fluid flows. *J Comput Phys* 2006;218:460–78. doi:10.1016/j.jcp.2006.02.017.
- [90] Sui Y, Chew Y, Roy P, Low H. A hybrid immersed-• boundary and multi-• block lattice Boltzmann method for simulating fluid and moving-• boundaries interactions. ... *Numer Methods Fluids* 2007;53:1727–54. doi:10.1002/fld.

- [91] Dupuis A, Chatelain P, Koumoutsakos P. An immersed boundary–lattice-Boltzmann method for the simulation of the flow past an impulsively started cylinder. *J Comput Phys* 2008;227:4486–98. doi:10.1016/j.jcp.2008.01.009.
- [92] Lu J, Han H, Shi B, Guo Z. Immersed boundary lattice Boltzmann model based on multiple relaxation times. *Phys Rev E* 2012;85:16711. doi:10.1103/PhysRevE.85.016711.
- [93] Suzuki K, Inamuro T. A higher-order immersed boundary-lattice Boltzmann method using a smooth velocity field near boundaries. *Comput Fluids* 2013;76:105–15. doi:10.1016/j.compfluid.2013.01.029.
- [94] Farnoush S, Manzari MT. Comparison of boundary slip for two variants of immersed boundary method in lattice Boltzmann framework. *Phys A Stat Mech Its Appl* 2014;404:200–16. doi:10.1016/j.physa.2014.02.010.
- [95] De Rosis A, Falcucci G, Ubertini S, Ubertini F. Aeroelastic study of flexible flapping wings by a coupled lattice Boltzmann-finite element approach with immersed boundary method. *J Fluids Struct* 2014;49:516–33. doi:10.1016/j.jfluidstructs.2014.05.010.
- [96] De Rosis A, Ubertini S, Ubertini F. A Comparison Between the Interpolated Bounce-Back Scheme and the Immersed Boundary Method to Treat Solid Boundary Conditions for Laminar Flows in the Lattice Boltzmann Framework. *J Sci Comput* 2014. doi:10.1007/s10915-014-9834-0.
- [97] De Rosis A, Ubertini S, Ubertini F. A partitioned approach for two-dimensional fluid-structure interaction problems by a coupled lattice Boltzmann-finite element method with immersed boundary. *J Fluids Struct* 2014;45:202–15. doi:10.1016/j.jfluidstructs.2013.12.009.
- [98] Li Z, Favier J, D’Ortona U, Poncet S. An immersed boundary-lattice Boltzmann method for single- and multi-component fluid flows. *J Comput Phys* 2015;304:424–40. doi:10.1016/j.jcp.2015.10.026.
- [99] Pellerin N, Leclaire S, Reggio M. Equilibrium distributions for straight, curved, and immersed boundary conditions in the lattice Boltzmann method. *Comput Fluids* 2014:126–35. doi:http://dx.doi.org/10.1016/j.compfluid.2014.06.007.
- [100] FlowKit Ltd. Palabos 2016. <http://wiki.palabos.org/models:bc> (accessed September 12,

2016).

- [101] Ghia U, Ghia K., Shin C. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J Comput Phys* 1982;48:387–411. doi:10.1016/0021-9991(82)90058-4.
- [102] Erturk E, Corke TC, Gökçöl C. Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers. *Int J Numer Methods Fluids* 2005;48:747–74. doi:10.1002/fld.953.
- [103] Higuera FJ, Succi S. Simulating the flow around a circular cylinder with a lattice Boltzmann equation. *Europhys Lett* 1989;8:517.
- [104] Izham M, Fukui T, Morinishi K. Application of regularized lattice Boltzmann method for incompressible flow simulation at high Reynolds number and flow with curved boundary. *J Fluid Sci Technol* 2011;6:812–22. doi:10.1299/jfst.6.812.
- [105] Pellerin N, Leclaire S, Reggio M. An implementation of the Spalart–Allmaras turbulence model in a multi-domain lattice Boltzmann method for solving turbulent airfoil flows. *Comput Math with Appl* 2015;70:3001–18. doi:10.1016/j.camwa.2015.10.006.
- [106] Rohde M, Kandhai D, Derksen JJ, Akker HEA Van Den. A generic, mass conservative local grid refinement technique for lattice-Boltzmann schemes. *Int J Numer Methods Fluids* 2006:439–68.
- [107] Eitel-Amor G, Meinke M, Schröder W. A lattice-Boltzmann method with hierarchically refined meshes. *Comput Fluids* 2013;75:127–39. doi:10.1016/j.compfluid.2013.01.013.
- [108] Geier M, Greiner A, Korvink JG. Bubble functions for the lattice Boltzmann method and their application to grid refinement. *Eur Phys J Spec Top* 2009;171:173–9. doi:10.1140/epjst/e2009-01026-6.
- [109] Spalart PR, Allmaras SR. A one-equation turbulence model for aerodynamic flows. 30th Aerosp. Sci. Meet. Exhib., Reno, NV: AIAA; 1992, p. 1–22.
- [110] Lockard DP, Luo L, Milder SD, Singer BA. Evaluation of PowerFLOW for aerodynamic applications. *J Stat Phys* 2002;107:423–78. doi:https://doi.org/10.1023/A:1014539411062.
- [111] Mei R, Yu D, Shyy W, Luo L-S. Force evaluation in the lattice Boltzmann method involving

- curved geometry. *Phys Rev E* 2002;65:41203. doi:10.1103/PhysRevE.65.041203.
- [112] Pellerin N, Leclaire S, Reggio M. Solving incompressible fluid flows on unstructured meshes with the lattice Boltzmann flux solver. *Eng Appl Comput Fluid Mech* 2017;11:310–27. doi:10.1080/19942060.2017.1292410.
- [113] Peng G, Xi H, Duncan C, Chou S-H. Lattice Boltzmann method on irregular meshes. *Phys Rev E* 1998;58:R4124–7. doi:10.1103/PhysRevE.58.R4124.
- [114] Peng G, Xi H, Duncan C, Chou S-H. Finite volume scheme for the lattice Boltzmann method on unstructured meshes. *Phys Rev E* 1999;59:4675–82. doi:10.1103/PhysRevE.59.4675.
- [115] Zarghami A, Maghrebi MJ, Ghasemi J, Ubertini S. Lattice Boltzmann finite volume formulation with improved stability. *Commun Comput Phys* 2012;12:42–64. doi:10.4208/cicp.151210.140711a.
- [116] Li W, Luo L-S. Finite volume Lattice Boltzmann Method for Nearly Incompressible Flows on Arbitrary Unstructured Meshes. *Commun Comput Phys* 2016;20:301–24. doi:10.4208/cicp.211015.040316a.
- [117] Shu C, Wang Y, Teo CJ, Wu J. Development of lattice Boltzmann flux solver for simulation of incompressible flows. *Adv Appl Math Mech* 2014;6:436–60. doi:10.4208/aamm.2014.4.s2.
- [118] Wang Y, Yang L, Shu C. From lattice Boltzmann method to lattice Boltzmann flux solver. *Entropy* 2015;7713–35. doi:10.3390/e17117713.
- [119] Geuzaine C, Remacle J-F. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *Int J Numer Methods Eng* 2009;79:1309–31. doi:10.1002/nme.2579.
- [120] Stiebler M, Tölke J, Krafczyk M. Advection–diffusion lattice Boltzmann scheme for hierarchical grids. *Comput Math with Appl* 2008;55:1576–84. doi:10.1016/j.camwa.2007.08.024.
- [121] Succi S, Amati G, Benzi R. Challenges in lattice Boltzmann computing. *J Stat Phys* 1995;81:5–16. doi:10.1007/BF02179964.

ANNEXE A – LA MÉTHODE DE BOLTZMANN SUR RÉSEAU ET LES ÉQUATIONS DE NAVIER-STOKES

Dans cette annexe nous démontrons, au moyen d'une expansion Chapman-Enskog, que la LBM avec opérateur de collision de type BGK est équivalente à résoudre les équations de Navier-Stokes en régime faiblement compressible.

L'équation de Boltzmann à vitesses finies s'écrit comme suit:

$$\frac{\partial f_i(\mathbf{x}, t)}{\partial t} + \mathbf{c}_i \cdot \nabla f_i(\mathbf{x}, t) = \Omega_i(\mathbf{x}, t) , \quad (\text{A.1})$$

où $\mathbf{c}_i = \Delta \mathbf{x}_i / \Delta t$. Nous transformons cette équation en remplaçant les dérivées partielles par une dérivée totale, puis par une différence finie :

$$\frac{f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t)}{\Delta t} = \Omega_i(\mathbf{x}, t) \quad (\text{A.2})$$

ou

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = \Delta t \Omega_i(\mathbf{x}, t) . \quad (\text{A.3})$$

A.1 Expansion en série de Taylor

Les distributions de probabilités sont tout d'abord réécrites en séries de Taylor, autour du point de référence (\mathbf{x}, t) telles que :

$$\begin{aligned} f_i(\mathbf{x}^*, t^*) \approx & f_i(\mathbf{x}, t) + \frac{\partial f_i(\mathbf{x}, t)}{\partial \mathbf{x}} \Delta \mathbf{x} + \frac{\partial f_i(\mathbf{x}, t)}{\partial t} \Delta t \\ & + \frac{1}{2} \left[\frac{\partial^2 f_i(\mathbf{x}, t)}{\partial \mathbf{x}^2} \Delta \mathbf{x}^2 + 2 \frac{\partial^2 f_i(\mathbf{x}, t)}{\partial \mathbf{x} \partial t} \Delta \mathbf{x} \Delta t + \frac{\partial^2 f_i(\mathbf{x}, t)}{\partial t^2} \Delta t^2 \right] , \end{aligned} \quad (\text{A.4})$$

que nous évaluons au point $(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t)$:

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) \approx f_i(\mathbf{x}, t) + \frac{\partial f_i(\mathbf{x}, t)}{\partial \mathbf{x}} \mathbf{c}_i \Delta t + \frac{\partial f_i(\mathbf{x}, t)}{\partial t} \Delta t + \frac{1}{2} \left[\frac{\partial^2 f_i(\mathbf{x}, t)}{\partial \mathbf{x}^2} (\mathbf{c}_i \Delta t)^2 + 2 \frac{\partial^2 f_i(\mathbf{x}, t)}{\partial \mathbf{x} \partial t} (\mathbf{c}_i \Delta t) \Delta t + \frac{\partial^2 f_i(\mathbf{x}, t)}{\partial t^2} \Delta t^2 \right]. \quad (\text{A.5})$$

Pour simplifier la suite des dérivations, nous omettrons (\mathbf{x}, t) et utiliserons la notation ∇ pour les dérivées partielles spatiales. En insérant (A.5) dans (A.3) nous obtenons donc la forme suivante de l'équation de Boltzmann :

$$\Delta t \left(\mathbf{c}_i \cdot \nabla + \frac{\partial}{\partial t} \right) f_i + \frac{1}{2} \Delta t^2 \left[(\mathbf{c}_i \cdot \nabla)^2 + 2 \frac{\partial}{\partial t} (\mathbf{c}_i \cdot \nabla) + \frac{\partial^2}{\partial t^2} \right] f_i = \Delta t \Omega_i \quad (\text{A.6})$$

ou

$$\left(\mathbf{c}_i \cdot \nabla + \frac{\partial}{\partial t} \right) f_i + \frac{1}{2} \Delta t \left[(\mathbf{c}_i \cdot \nabla)^2 + 2 \frac{\partial}{\partial t} (\mathbf{c}_i \cdot \nabla) + \frac{\partial^2}{\partial t^2} \right] f_i = \Omega_i. \quad (\text{A.7})$$

A.2 Expansion Chapman-Enskog

L'expansion Chapman-Enskog consiste à séparer les variables constituantes de l'équation en une série d'échelles. Les échelles temporelles peuvent être associées à des phénomènes physiques se produisant à des vitesses différentes. Ici, t_1 représente l'advection, qui se produit plus rapidement que la diffusion, représentée par t_2 .

$$\frac{\partial}{\partial t} = \varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2} + O(\varepsilon^3) \quad (\text{A.8})$$

$$\nabla = \varepsilon \nabla_1 + O(\varepsilon^2) \quad (\text{A.9})$$

$$f_i = f_i^{(0)} + \varepsilon f_i^{(1)} + O(\varepsilon^2) \quad (\text{A.10})$$

$$\Omega_i = \Omega_i^{(0)} + \varepsilon \Omega_i^{(1)} + \varepsilon^2 \Omega_i^{(2)} + O(\varepsilon^3) \quad (\text{A.11})$$

L'Éq. (A.7) est réécrite en utilisant les relations (A.8) à (A.11) :

$$\begin{aligned}
\Omega_i &= \Omega_i^{(0)} + \varepsilon \Omega_i^{(1)} + \varepsilon^2 \Omega_i^{(2)} \\
&= \left(\mathbf{c}_i \cdot \varepsilon \nabla_1 + \varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2} \right) (f_i^{(0)} + \varepsilon f_i^{(1)}) \\
&\quad + \frac{1}{2} \Delta t \left((\mathbf{c}_i \cdot \varepsilon \nabla_1)^2 + 2 \left(\varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2} \right) (\mathbf{c}_i \cdot \varepsilon \nabla_1) + \left(\varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2} \right)^2 \right) (f_i^{(0)} + \varepsilon f_i^{(1)})
\end{aligned} \tag{1.12}$$

que nous séparons ensuite selon les échelles choisies en omettant les termes d'ordre ε^3 et plus.

$$\varepsilon^0 : \Omega_i^{(0)} = 0 \tag{A.13}$$

$$\varepsilon^1 : \Omega_i^{(1)} = \left(\mathbf{c}_i \cdot \nabla_1 + \frac{\partial}{\partial t_1} \right) f_i^{(0)} \tag{A.14}$$

$$\varepsilon^2 : \Omega_i^{(2)} = \left(\frac{\partial}{\partial t_2} + \frac{1}{2} \Delta t \left((\mathbf{c}_i \cdot \nabla_1)^2 + 2 \frac{\partial}{\partial t_1} (\mathbf{c}_i \cdot \nabla_1) + \frac{\partial^2}{\partial t_1^2} \right) \right) f_i^{(0)} + \left(\mathbf{c}_i \cdot \nabla_1 + \frac{\partial}{\partial t_1} \right) f_i^{(1)} \tag{A.15}$$

A.3 Moments des distributions de probabilités

Les quantités hydrodynamiques macroscopiques, telles que la densité et la quantité de mouvement (par unité de volume), sont définies comme les moments des distributions de probabilités par les vitesses discrètes du réseau. Si l'on omet les forces externes ou les termes sources, ces quantités macroscopiques doivent être conservées, ce qui est reflété à l'échelle mésoscopique par l'égalité des moments d'ordre 0 et d'ordre 1 des distributions de probabilités à l'équilibre et non-équilibrées.

$$\rho = \sum_i f_i = \sum_i f_i^{(0)} \tag{A.16}$$

$$\rho \mathbf{u} = \sum_i \mathbf{c}_i f_i = \sum_i \mathbf{c}_i f_i^{(0)} \tag{A.17}$$

Les Éqns. (A.10), (A.16) et (A.17) permettent de déduire que les moments d'ordre 0 et d'ordre 1 des distributions de probabilités de rang 1 et plus sont nulles.

$$\left. \begin{aligned} \sum_i f_i^{(k)} &= 0 \\ \sum_i \mathbf{c}_i f_i^{(k)} &= 0 \end{aligned} \right\} k > 0 \tag{A.18}$$

Il est à noter que les moments supérieurs à 1 ne sont pas nécessairement conservés. Puisque les moments d'ordre 0 et d'ordre 1 des distributions de probabilités à l'équilibre et non-équilibrées sont égaux, et que la relation entre ces distributions dépend du terme de collision Ω , les moments de ce terme de collision doivent aussi être nuls.

$$\sum_i \Omega_i^{(0)} = 0, \quad \sum_i \Omega_i^{(1)} = 0, \quad \sum_i \Omega_i^{(2)} = 0 \quad (\text{A.19})$$

$$\sum_i \mathbf{c}_i \Omega_i^{(0)} = 0, \quad \sum_i \mathbf{c}_i \Omega_i^{(1)} = 0, \quad \sum_i \mathbf{c}_i \Omega_i^{(2)} = 0 \quad (\text{A.20})$$

Nous utilisons maintenant les propriétés énoncées par les Éqns. (A.16) à (A.20) pour appliquer des moments d'ordre 0 et d'ordre 1 aux Éqns. (A.13) à (A.15).

A.4 Moments d'ordre 0

Éqn. (A.14) :

$$\begin{aligned} \sum_i \Omega_i^{(1)} &= \sum_i \left(\mathbf{c}_i \cdot \nabla_1 + \frac{\partial}{\partial t_1} \right) f_i^{(0)}, \\ 0 &= \nabla_1 \cdot \sum_i \mathbf{c}_i f_i^{(0)} + \frac{\partial}{\partial t_1} \sum_i f_i^{(0)}, \\ 0 &= \nabla_1 \cdot (\rho \mathbf{u}) + \frac{\partial \rho}{\partial t_1}. \end{aligned} \quad (\text{A.21})$$

Éqn. (A.15) :

$$\begin{aligned} \sum_i \Omega_i^{(2)} &= \sum_i \left(\frac{\partial}{\partial t_2} + \frac{1}{2} \Delta t \left((\mathbf{c}_i \cdot \nabla_1)^2 + 2 \frac{\partial}{\partial t_1} (\mathbf{c}_i \cdot \nabla_1) + \frac{\partial^2}{\partial t_1^2} \right) \right) f_i^{(0)} + \sum_i \left(\mathbf{c}_i \cdot \nabla_1 + \frac{\partial}{\partial t_1} \right) f_i^{(1)}, \\ 0 &= \frac{\partial}{\partial t_2} \sum_i f_i^{(0)} + \frac{1}{2} \Delta t \nabla_1 \nabla_1^T : \sum_i \mathbf{c}_i \mathbf{c}_i^T f_i^{(0)} + \Delta t \frac{\partial}{\partial t_1} \nabla_1 \cdot \sum_i \mathbf{c}_i f_i^{(0)} \\ &\quad + \frac{1}{2} \Delta t \frac{\partial^2}{\partial t_1^2} \sum_i f_i^{(0)} + \nabla_1 \cdot \sum_i \mathbf{c}_i f_i^{(1)} + \frac{\partial}{\partial t_1} \sum_i f_i^{(1)}, \\ 0 &= \frac{\partial \rho}{\partial t_2} + \frac{1}{2} \Delta t \nabla_1 \nabla_1^T : \mathbf{T}^{(0)} + \Delta t \frac{\partial}{\partial t_1} \nabla_1 \cdot (\rho \mathbf{u}) + \frac{1}{2} \Delta t \frac{\partial^2 \rho}{\partial t_1^2}, \end{aligned} \quad (\text{A.22})$$

où $\mathbf{T}^{(0)} = \sum_i \mathbf{c}_i \mathbf{c}_i^T f_i^{(0)}$ est un tenseur d'ordre 2.

On soustrait Δt multiplié par la demie de la dérivée en t_1 de (A.21) à (A.22),

$$0 = \frac{\partial \rho}{\partial t_2} + \frac{1}{2} \Delta t \left(\nabla_1 \nabla_1^T : \mathbf{T}^{(0)} + \frac{\partial}{\partial t_1} \nabla_1 \cdot (\rho \mathbf{u}) \right). \quad (\text{A.23})$$

A.5 Moments d'ordre 1

Éqn. (A.14) :

$$\begin{aligned} \sum_i \mathbf{c}_i \Omega_i^{(1)} &= \sum_i \mathbf{c}_i \left(\mathbf{c}_i \cdot \nabla_1 + \frac{\partial}{\partial t_1} \right) f_i^{(0)}, \\ 0 &= \nabla_1^T \cdot \sum_i \mathbf{c}_i \mathbf{c}_i^T f_i^{(0)} + \frac{\partial}{\partial t_1} \sum_i \mathbf{c}_i f_i^{(0)}, \\ 0 &= \nabla_1^T \cdot \mathbf{T}^{(0)} + \frac{\partial}{\partial t_1} (\rho \mathbf{u}). \end{aligned} \quad (\text{A.24})$$

Dans l'Éqn. (A.24) nous avons utilisé la notation suivante

$$\nabla_1^T \cdot \mathbf{T}^{(0)} \equiv \begin{bmatrix} \nabla_{1x} & \nabla_{1y} \end{bmatrix} \cdot \begin{bmatrix} T_{11}^{(0)} & T_{12}^{(0)} \\ T_{21}^{(0)} & T_{22}^{(0)} \end{bmatrix} = \begin{bmatrix} \nabla_{1x} T_{11}^{(0)} + \nabla_{1y} T_{12}^{(0)} \\ \nabla_{1x} T_{21}^{(0)} + \nabla_{1y} T_{22}^{(0)} \end{bmatrix}. \quad (\text{A.25})$$

En multipliant l'Éqn. (A.24) par $\frac{1}{2} \Delta t \nabla_1^T$ puis en l'insérant dans l'éqn. (A.23), celle-ci est réduite à :

$$\frac{\partial \rho}{\partial t_2} = 0. \quad (\text{A.26})$$

Éqn. (A.15) :

$$\sum_i \mathbf{c}_i \Omega_i^{(2)} = \sum_i \mathbf{c}_i \left(\frac{\partial}{\partial t_2} + \frac{1}{2} \Delta t \left((\mathbf{c}_i \cdot \nabla_1)^2 + 2 \frac{\partial}{\partial t_1} \mathbf{c}_i \cdot \nabla_1 + \frac{\partial^2}{\partial t_1^2} \right) \right) f_i^{(0)} + \sum_i \mathbf{c}_i \left(\mathbf{c}_i \cdot \nabla_1 + \frac{\partial}{\partial t_1} \right) f_i^{(1)},$$

$$\begin{aligned}
0 &= \frac{\partial}{\partial t_2} \sum_i \mathbf{c}_i f_i^{(0)} + \frac{1}{2} \Delta t \sum_i \mathbf{c}_i (\mathbf{c}_i \cdot \nabla_1)^2 f_i^{(0)} + \Delta t \frac{\partial}{\partial t_1} \sum_i \mathbf{c}_i (\mathbf{c}_i \cdot \nabla_1) f_i^{(0)} \\
&\quad + \frac{1}{2} \Delta t \frac{\partial^2}{\partial t_1^2} \sum_i \mathbf{c}_i f_i^{(0)} + \sum_i \mathbf{c}_i (\mathbf{c}_i \cdot \nabla_1) f_i^{(1)} + \frac{\partial}{\partial t_1} \sum_i \mathbf{c}_i f_i^{(1)} , \\
0 &= \frac{\partial \rho \mathbf{u}}{\partial t_2} + \frac{1}{2} \Delta t \nabla_1 \nabla_1^T : \sum_i \mathbf{c}_i (\mathbf{c}_i \mathbf{c}_i^T) f_i^{(0)} + \Delta t \frac{\partial}{\partial t_1} \nabla_1^T \cdot \sum_i \mathbf{c}_i \mathbf{c}_i^T f_i^{(0)} \\
&\quad + \frac{1}{2} \Delta t \frac{\partial^2 \rho \mathbf{u}}{\partial t_1^2} + \nabla_1^T \cdot \sum_i \mathbf{c}_i \mathbf{c}_i^T f_i^{(1)} , \\
0 &= \frac{\partial \rho \mathbf{u}}{\partial t_2} + \nabla_1^T \cdot \mathbf{T}^{(1)} + \frac{1}{2} \Delta t \left(\nabla_1 \nabla_1^T : \mathbf{R}^{(0)} + 2 \frac{\partial}{\partial t_1} \nabla_1^T \cdot \mathbf{T}^{(0)} + \frac{\partial^2 \rho \mathbf{u}}{\partial t_1^2} \right) , \tag{A.27}
\end{aligned}$$

où $\mathbf{R}^{(0)} = \sum_i \mathbf{c}_i (\mathbf{c}_i \mathbf{c}_i^T) f_i^{(0)}$ est une combinaison de deux tenseurs d'ordre 2, car nous effectuons les dérivations pour un système bidimensionnel.

A.6 Équations hydrodynamiques

A.6.1 Continuité

L'équation de continuité (conservation de masse) est construite au moyen des Éqns. (A.8), (A.9), (A.21) et (A.26). Tout d'abord, on multiplie (A.21) par ε et (A.26) par ε^2 , puis on additionne :

$$\begin{aligned}
\varepsilon \frac{\partial \rho}{\partial t_1} + \varepsilon \nabla_1 \cdot (\rho \mathbf{u}) + \varepsilon^2 \frac{\partial \rho}{\partial t_2} &= 0 , \\
\left(\varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2} \right) \rho + \varepsilon \nabla_1 \cdot (\rho \mathbf{u}) &= 0 . \tag{A.28}
\end{aligned}$$

Ensuite, on utilise les définitions de (A.8) et (A.9) pour simplifier (A.28) et ainsi obtenir l'équation de continuité.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{A.29}$$

A.6.2 Quantité de mouvement

Les équations de conservation de quantité de mouvement sont construites au moyen des Éqns. (A.8), (A.9), (A.24) et (A.27). Tout d'abord, on soustrait Δt multiplié par la demie de la dérivée en t_1 de (A.24) à (A.27) :

$$0 = \frac{\partial \rho \mathbf{u}}{\partial t_2} + \nabla_1^T \cdot \mathbf{T}^{(1)} + \frac{1}{2} \Delta t \left(\nabla_1 \nabla_1^T : \mathbf{R}^{(0)} + 2 \frac{\partial}{\partial t_1} \nabla_1^T \cdot \mathbf{T}^{(0)} + \frac{\partial^2 \rho \mathbf{u}}{\partial t_1^2} \right) - \frac{1}{2} \Delta t \frac{\partial}{\partial t_1} \left(\nabla_1^T \cdot \mathbf{T}^{(0)} + \frac{\partial \rho \mathbf{u}}{\partial t_1} \right),$$

$$0 = \frac{\partial \rho \mathbf{u}}{\partial t_2} + \nabla_1^T \cdot \mathbf{T}^{(1)} + \frac{1}{2} \Delta t \left(\nabla_1 \nabla_1^T : \mathbf{R}^{(0)} + \frac{\partial}{\partial t_1} \nabla_1^T \cdot \mathbf{T}^{(0)} \right). \quad (\text{A.30})$$

Ensuite, on multiplie (A.24) par ε et (A.30) par ε^2 , puis on additionne :

$$\varepsilon \left(\nabla_1^T \cdot \mathbf{T}^{(0)} + \frac{\partial \rho \mathbf{u}}{\partial t_1} \right) + \varepsilon^2 \left(\frac{\partial \rho \mathbf{u}}{\partial t_2} + \nabla_1^T \cdot \mathbf{T}^{(1)} + \frac{1}{2} \Delta t \left(\nabla_1 \nabla_1^T : \mathbf{R}^{(0)} + \frac{\partial}{\partial t_1} \nabla_1^T \cdot \mathbf{T}^{(0)} \right) \right) = 0,$$

$$\varepsilon \left(\frac{\partial}{\partial t_1} + \varepsilon \frac{\partial}{\partial t_2} \right) \rho \mathbf{u} + \varepsilon \nabla_1^T \cdot (\mathbf{T}^{(0)} + \varepsilon \mathbf{T}^{(1)}) + \frac{1}{2} \Delta t \varepsilon^2 \left(\nabla_1 \nabla_1^T : \mathbf{R}^{(0)} + \frac{\partial}{\partial t_1} \nabla_1^T \cdot \mathbf{T}^{(0)} \right) = 0. \quad (\text{A.31})$$

En utilisant les définitions de (A.8) et (A.9), on simplifie (A.31) :

$$\varepsilon \left(\nabla_1^T \cdot \mathbf{T}^{(0)} + \frac{\partial \rho \mathbf{u}}{\partial t_1} \right) + \varepsilon^2 \left(\frac{\partial \rho \mathbf{u}}{\partial t_2} + \nabla_1^T \cdot \mathbf{T}^{(1)} + \frac{1}{2} \Delta t \left(\nabla_1 \nabla_1^T : \mathbf{R}^{(0)} + \frac{\partial}{\partial t_1} \nabla_1^T \cdot \mathbf{T}^{(0)} \right) \right) = 0,$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla^T \cdot (\mathbf{T}^{(0)} + \varepsilon \mathbf{T}^{(1)}) + \frac{1}{2} \Delta t \varepsilon \frac{\partial}{\partial t_1} \nabla^T \cdot \mathbf{T}^{(0)} + \frac{1}{2} \Delta t \varepsilon \nabla \nabla_1^T : \mathbf{R}^{(0)} = 0. \quad (\text{A.32})$$

Afin d'obtenir les équations de Navier-Stokes, nous devons résoudre les termes $\mathbf{T}^{(0)}$, $\mathbf{T}^{(1)}$ et $\mathbf{R}^{(0)}$, ainsi que leurs produits avec l'opérateur ∇ . Les termes $\mathbf{T}^{(0)}$ et $\mathbf{R}^{(0)}$ peuvent être obtenus de façon explicite au moyen des distributions à l'équilibre choisies. Le terme $\mathbf{T}^{(1)}$ est aussi obtenu au moyen de ces distributions mais doit au préalable être reformulé. Pour ce faire, nous employons le modèle de collision BGK, ainsi que les Éqns. (A.10), (A.11), (A.14) et (A.19) :

$$\Omega_i = \frac{1}{\Delta t_R} (f_i^{(0)} - f_i),$$

$$\Omega_i = \Omega_i^{(0)} + \varepsilon \Omega_i^{(1)} + \dots = \frac{1}{\Delta t_R} \left(f_i^{(0)} - \left(f_i^{(0)} + \varepsilon f_i^{(1)} + \dots \right) \right),$$

$$\Omega_i^{(1)} = \frac{-1}{\Delta t_R} f_i^{(1)} = \left(\mathbf{c}_i \cdot \nabla_1 + \frac{\partial}{\partial t_1} \right) f_i^{(0)}. \quad (\text{A.33})$$

Nous obtenons donc $f_i^{(1)}$ en fonction de $f_i^{(0)}$:

$$f_i^{(1)} = -\Delta t_R \left(\mathbf{c}_i \cdot \nabla_1 + \frac{\partial}{\partial t_1} \right) f_i^{(0)}, \quad (\text{A.34})$$

ce qui permettra de résoudre $\mathbf{T}^{(1)}$ au moyen des distributions à l'équilibre.

A.6.3 Distributions à l'équilibre

Le modèle de vitesses de réseau D2Q9 et les distributions à l'équilibre suivantes, présentées à la section 1.1.4, permettent de retrouver les équations de Navier-Stokes.

$$f_i^{(0)} = \rho w_i \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{1}{2} \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{c_s^4} - \frac{1}{2} \frac{\mathbf{u} \cdot \mathbf{u}}{c_s^2} \right) \quad (\text{A.35})$$

$$\mathbf{w} = (16, 4, 4, 4, 4, 1, 1, 1, 1)/36 \quad (\text{A.36})$$

$$\mathbf{c}_i = (c_x, c_y)_i = \begin{cases} (0, 0) & i = 1 \\ c \left(\cos[\pi(i-2)/2], \sin[\pi(i-2)/2] \right) & i = 2, 3, 4, 5 \\ c \left(\cos[\pi(i-6)/2 + \pi/4], \sin[\pi(i-6)/2 + \pi/4] \right) & i = 6, 7, 8, 9 \end{cases} \quad (\text{A.37})$$

$$c = \frac{\Delta x}{\Delta t}, \quad c_s = \frac{c}{\sqrt{3}} \quad (\text{A.38})$$

A.6.4 Terme $\mathbf{T}^{(0)}$

Rappelons-nous la définition de $\mathbf{T}^{(0)}$, $\mathbf{T}^{(0)} = \sum_i (\mathbf{c}_i \mathbf{c}_i^T) f_i^{(0)}$. Nous résolvons ce terme au moyen d'un logiciel de calcul symbolique, en omettant de reproduire ici toutes les dérivations algébriques :

$$\mathbf{T}^{(0)} = \rho \begin{bmatrix} c_s^2 + u_x^2 & u_x u_y \\ u_x u_y & c_s^2 + u_y^2 \end{bmatrix}. \quad (\text{A.39})$$

Nous réécrivons l'Éqn. (A.39) sous une forme vectorielle simplifiée :

$$\mathbf{T}^{(0)} = \rho c_s^2 \mathbf{I} + \rho \mathbf{u} \mathbf{u}^T. \quad (\text{A.40})$$

A.6.5 Terme $\mathbf{R}^{(0)}$

Rappelons-nous la définition de $\mathbf{R}^{(0)}$, $\mathbf{R}^{(0)} = \sum \mathbf{c}_i (\mathbf{c}_i \mathbf{c}_i^T) f_i^{(0)}$. Nous résolvons aussi ce terme au moyen d'un logiciel de calcul symbolique :

$$\mathbf{R}^{(0)} = \begin{matrix} \rho c_s^2 \begin{bmatrix} 3u_x & u_y \\ u_y & u_x \end{bmatrix} \\ \rho c_s^2 \begin{bmatrix} u_y & u_x \\ u_x & 3u_y \end{bmatrix} \end{matrix}. \quad (\text{A.41})$$

A.6.6 Terme $\mathbf{T}^{(1)}$

Rappelons-nous la définition de $\mathbf{T}^{(1)}$, $\mathbf{T}^{(1)} = \sum_i \mathbf{c}_i \mathbf{c}_i^T f_i^{(1)}$. Après les dérivations algébriques, ce terme devient :

$$\mathbf{T}^{(1)} = -\Delta t_R \begin{bmatrix} c_s^2 (\nabla_{1x} \rho u_x + \nabla_{1y} \rho u_y) + 2c_s^2 \nabla_{1x} \rho u_x & c_s^2 (\nabla_{1x} \rho u_y + \nabla_{1y} \rho u_x) + \frac{\partial \rho u_x u_y}{\partial t_1} \\ + \frac{\partial \rho u_x^2}{\partial t_1} + c_s^2 \frac{\partial \rho}{\partial t_1} & \\ c_s^2 (\nabla_{1x} \rho u_y + \nabla_{1y} \rho u_x) + \frac{\partial \rho u_x u_y}{\partial t_1} & c_s^2 (\nabla_{1x} \rho u_x + \nabla_{1y} \rho u_y) + 2c_s^2 \nabla_{1y} \rho u_y \\ & + \frac{\partial \rho u_y^2}{\partial t_1} + c_s^2 \frac{\partial \rho}{\partial t_1} \end{bmatrix}. \quad (\text{A.42})$$

Nous réécrivons l'Éqn. (A.42) sous une forme vectorielle simplifiée :

$$\mathbf{T}^{(1)} = -\Delta t_R c_s^2 \left(\frac{\partial \rho}{\partial t_1} + \nabla_1 \cdot \rho \mathbf{u} \right) \mathbf{I} - \Delta t_R \frac{\partial}{\partial t_1} (\rho \mathbf{u} \mathbf{u}^T) - \Delta t_R c_s^2 \left(\left(\nabla_1 (\rho \mathbf{u}^T) \right) + \left(\nabla_1 (\rho \mathbf{u}^T) \right)^T \right). \quad (\text{A.43})$$

Le premier terme du côté droit de l'Éqn. (A.43) est réduit à 0 par l'Éqn. (A.21) :

$$\mathbf{T}^{(1)} = -\Delta t_R \frac{\partial}{\partial t_1} (\rho \mathbf{u} \mathbf{u}^T) - \Delta t_R c_s^2 \left(\left(\nabla_1 (\rho \mathbf{u}^T) \right) + \left(\nabla_1 (\rho \mathbf{u}^T) \right)^T \right). \quad (\text{A.44})$$

A.6.7 Terme $\nabla \nabla_1^T : \mathbf{R}^{(0)}$

Sans perte de généralité, nous pouvons dériver algébriquement $\nabla \nabla^T : \mathbf{R}^{(0)}$, qui devient :

$$\nabla \nabla^T : \mathbf{R}^{(0)} = c_s^2 \begin{bmatrix} 3 \frac{\partial^2 \rho u_x}{\partial x^2} + 2 \frac{\partial^2 \rho u_y}{\partial x \partial y} + \frac{\partial^2 \rho u_x}{\partial y^2} \\ 3 \frac{\partial^2 \rho u_y}{\partial x^2} + 2 \frac{\partial^2 \rho u_x}{\partial x \partial y} + 3 \frac{\partial^2 \rho u_y}{\partial y^2} \end{bmatrix}, \quad (\text{A.45})$$

puis le réécrire sous une forme vectorielle simplifiée :

$$\nabla \nabla^T : \mathbf{R}^{(0)} = c_s^2 \left(2 \nabla (\nabla \cdot \rho \mathbf{u}) + (\nabla \cdot \nabla) \rho \mathbf{u} \right). \quad (\text{A.46})$$

Par une simple analogie, le terme $\nabla \nabla_1^T : \mathbf{R}^{(0)}$ est obtenu :

$$\nabla \nabla_1^T : \mathbf{R}^{(0)} = c_s^2 \left(2 \nabla (\nabla_1 \cdot \rho \mathbf{u}) + (\nabla \cdot \nabla_1) \rho \mathbf{u} \right). \quad (\text{A.47})$$

A.6.8 Terme $\frac{\partial}{\partial t_1} \nabla^T \cdot \mathbf{T}^{(0)}$

Après les dérivations algébriques, ce terme devient :

$$\frac{\partial}{\partial t_1} \nabla^T \cdot \mathbf{T}^{(0)} = c_s^2 \frac{\partial}{\partial t_1} \begin{bmatrix} \frac{\partial \rho}{\partial x} \\ \frac{\partial \rho}{\partial y} \end{bmatrix} + \frac{\partial}{\partial t_1} \begin{bmatrix} \frac{\partial \rho u_x^2}{\partial x} + \frac{\partial \rho u_x u_y}{\partial y} \\ \frac{\partial \rho u_x u_y}{\partial x} + \frac{\partial \rho u_y^2}{\partial y} \end{bmatrix}. \quad (\text{A.48})$$

On ramène l'Éqn. (A.48) à une formulation vectorielle :

$$\frac{\partial}{\partial t_1} \nabla^T \cdot \mathbf{T}^{(0)} = c_s^2 \nabla (-\nabla_1 \cdot \rho \mathbf{u}) + \frac{\partial}{\partial t_1} \nabla^T \cdot (\rho \mathbf{u} \mathbf{u}^T). \quad (\text{A.49})$$

A.6.9 Combinaison des termes

Nous remplaçons maintenant les termes dans l'Éqn. (A.32) :

$$\begin{aligned}
& \frac{\partial \rho \mathbf{u}}{\partial t} + \nabla^T \cdot \left(\rho c_s^2 \mathbf{I} + \rho \mathbf{u} \mathbf{u}^T + \varepsilon \left(-\Delta t_R \frac{\partial}{\partial t_1} (\rho \mathbf{u} \mathbf{u}^T) - \Delta t_R c_s^2 \left((\nabla_1 (\rho \mathbf{u}^T)) + (\nabla_1 (\rho \mathbf{u}^T))^T \right) \right) \right) \\
& + \frac{1}{2} \Delta t \varepsilon \left(c_s^2 \nabla (-\nabla_1 \cdot \rho \mathbf{u}) + \frac{\partial}{\partial t_1} \nabla^T \cdot (\rho \mathbf{u} \mathbf{u}^T) \right) \\
& + \frac{1}{2} \Delta t \varepsilon \left(c_s^2 (2 \nabla (\nabla_1 \cdot \rho \mathbf{u}) + (\nabla \cdot \nabla_1) \rho \mathbf{u}) \right) \quad . \quad (\text{A.50}) \\
& = 0
\end{aligned}$$

En utilisant (A.9), les termes $\varepsilon \nabla_1$ sont ramenés à ∇ , et après simplification l'Éqn. (A.50) est réduite à :

$$\begin{aligned}
& \frac{\partial \rho \mathbf{u}}{\partial t} + \nabla^T \cdot (\rho c_s^2 \mathbf{I} + \rho \mathbf{u} \mathbf{u}^T) \\
& - c_s^2 \left(\Delta t_R - \frac{1}{2} \Delta t \right) \times \left(\nabla (\nabla \cdot \rho \mathbf{u}) + (\nabla \cdot \nabla) \rho \mathbf{u} + \frac{\varepsilon}{c_s^2} \frac{\partial}{\partial t_1} (\nabla^T \cdot (\rho \mathbf{u} \mathbf{u}^T)) \right) = 0 \quad . \quad (\text{A.51})
\end{aligned}$$

En assumant un écoulement faiblement compressible, avec $Ma = |\mathbf{u}| / c_s \ll 1$, le dernier terme de l'Éqn. (A.51) devient négligeable.

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla^T \cdot (\rho \mathbf{u} \mathbf{u}^T) = -\nabla^T \cdot (\rho c_s^2 \mathbf{I}) + c_s^2 \Delta t \left(\frac{\Delta t_R}{\Delta t} - \frac{1}{2} \right) (\nabla (\nabla \cdot \rho \mathbf{u}) + (\nabla \cdot \nabla) \rho \mathbf{u}) \quad . \quad (\text{A.52})$$

Finalement, les équations de Navier-Stokes sont obtenues:

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla^T \cdot (\rho \mathbf{u} \mathbf{u}^T) = -\nabla^T \cdot (p \mathbf{I}) + \nu (\nabla (\nabla \cdot \rho \mathbf{u}) + (\nabla \cdot \nabla) \rho \mathbf{u}) \quad , \quad (\text{A.53})$$

où

$$p = \rho c_s^2 \quad , \quad \nu = c_s^2 \Delta t \left(\frac{\Delta t_R}{\Delta t} - \frac{1}{2} \right) \quad . \quad (\text{A.54})$$

On remarque que dans l'éqn. (A.53), le terme impliquant la seconde viscosité est absent, ce qui implique la condition d'incompressibilité $\nabla \cdot \mathbf{u} = 0$. Or, la relation $p = \rho c_s^2$ indique que la pression est proportionnelle à la densité, indiquant que cette dernière n'est pas constante. Cette contradiction apparente nous montre plutôt que la méthode de Boltzmann sur réseau correspond en réalité à résoudre les équations différentielles pour un écoulement faiblement compressible, tel que nous l'avons assumé pour obtenir l'Éqn. (A.52).

Par ailleurs, les équations de Navier-Stokes peuvent être écrites de façon simplifiée en ayant recours au tenseur de flux de quantité de mouvement Π :

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla^T \cdot \Pi = 0 . \quad (\text{A.55})$$

Or, par manipulation algébrique, on constate que l'Éqn. (A.50) peut être réécrite en fonction des tenseurs $\mathbf{T}^{(0)}$ et $\mathbf{T}^{(1)}$,

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla^T \cdot \left(\mathbf{T}^{(0)} + \varepsilon \left(1 - \frac{1}{2} \frac{\Delta t}{\Delta t_R} \right) \mathbf{T}^{(1)} \right) = 0 . \quad (\text{A.56})$$

Donc, pour une approximation tronquée à l'échelle ε^2 , nous obtenons l'équation suivante :

$$\Pi = \sum_{i=1}^N (\mathbf{c}_i \mathbf{c}_i^T) \left(f_i^{eq} + \left(1 - \frac{1}{2} \frac{\Delta t}{\Delta t_R} \right) f_i^{neq} \right) , \quad (\text{A.57})$$

où $f_i^{eq} = f_i^{(0)}$ et $f_i^{neq} = f_i - f_i^{(0)} \approx \varepsilon f_i^{(1)}$. De la même façon, l'Éqn. (A.34) peut être approximée par :

$$f_i^{neq} = -\Delta t_R \left(\mathbf{c}_i \cdot \nabla + \frac{\partial}{\partial t} \right) f_i^{eq} . \quad (\text{A.58})$$

ANNEXE B – ARTICLE 1 : EQUILIBRIUM DISTRIBUTIONS FOR STRAIGHT, CURVED, AND IMMERSED BOUNDARY CONDITIONS IN THE LATTICE BOLTZMANN METHOD

Cet article a été publié dans une revue scientifique avec comité de lecture. La version publiée diffère de la version soumise suite à des modifications effectuées par l'éditeur et à des corrections que nous avons effectuées en ligne avec l'outil d'édition fourni par l'éditeur.

Pellerin N, Leclaire S, Reggio M. Equilibrium distributions for straight, curved, and immersed boundary conditions in the lattice Boltzmann method. *Comput Fluids* 2014;126-35.

Date: Jun 05, 2014
To: "Nicolas Pellerin" nicolas-3.pellerin@polymtl.ca,nicolas.pellerin@nrc-cnrc.gc.ca
From: "Computers and Fluids" CAF@elsevier.com
Subject: Your Submission
 Ms. Ref. No.: CAF-D-13-00534
 Title: Equilibrium Distributions for Straight, Curved, and Immersed Boundary Conditions in the Lattice Boltzmann Method
 Computers and Fluids

Dear Nicolas,

I am pleased to inform you that your paper
 "Equilibrium Distributions for Straight, Curved, and Immersed Boundary Conditions in the Lattice Boltzmann Method" has been accepted for publication in *Computers and Fluids*.

Thank you for submitting your work to *Computers and Fluids*.

Sincerely,

Michel Deville, Em. Professor
 Editor-in-Chief
 Computers and Fluids

Figure B.1 : Avis d'acceptation par les éditeurs de l'article «Equilibrium distributions for straight, curved, and immersed boundary conditions in the lattice Boltzmann method».

Title

Equilibrium Distributions for Straight, Curved, and Immersed Boundary Conditions in the Lattice Boltzmann Method

Author (corresponding)

Nicolas Pellerin

Department of Mechanical Engineering, École Polytechnique, 2500 chemin de Polytechnique,
Montréal, Québec H3T 1J4, Canada
nicolas-3.pellerin@polymtl.ca
(613) 297-6176

Co-authors

Sébastien Leclaire

Department of Mechanical Engineering, École Polytechnique, 2500 chemin de Polytechnique,
Montréal, Québec H3T 1J4, Canada
sebastien.leclaire@polymtl.ca
(514) 340-5121, ext. 5895

Marcelo Reggio

Department of Mechanical Engineering, École Polytechnique, 2500 chemin de Polytechnique,
Montréal, Québec H3T 1J4, Canada
marcelo.reggio@polymtl.ca
(514) 340-5121, ext. 4411

Abstract

A versatile velocity boundary condition treatment for the lattice Boltzmann method is proposed. The method can handle straight and curved boundaries, whether immersed or not. It is based on the streaming of equilibrium particle distributions emanating from ghost nodes. The fundamental idea is the location of these ghost nodes, specifically chosen on the lattice to avoid the evaluation of non equilibrium distributions. The methodology considers that the particle collision process time on the ghost nodes is proportional to the distance from the fluid nodes missing distributions. The linear nature of the BGK operator, which relaxes distributions exactly to their equilibrium state when the collision time is equal to the relaxation time, is also a key element. The method is compared to two other classical approaches for boundary condition treatment in the lattice Boltzmann method, on four test cases for which results are available in the scientific literature. Overall, the proposed method is second order accurate, and, when compared to the other two approaches, smaller errors were observed, especially for flows with a shorter relaxation time.

Keywords

Lattice Boltzmann, boundary condition, immersed boundary, curved boundary.

1. Introduction

In computational fluid dynamics, the boundary conditions have a strong impact on the numerical stability and accuracy of the simulated flow solutions. This general observation not only applies to traditional methods for solving the Navier-Stokes equations, such as finite difference or finite element schemes, but also to the lattice Boltzmann method (LBM), which does not treat the fluid as a continuum. This method deals with the dynamics of particles moving and colliding over a structured lattice, and from which macroscopic fluid properties are retrieved through moments using lattice speeds. A great feature of the LBM, owing to its application on a structured grid and the locality of the particle collision process, is that it is easy to parallelize. Another advantage is its ability to handle curved boundary conditions without conforming the lattice to the boundary geometry.

The first method for treating curved boundaries with the LBM was proposed by Filippova and Hänel [1]. They calculated the particle distribution coming from a boundary with a linear combination of the distributions on the fluid node neighboring the boundary and the equilibrium distributions on the solid node located on the opposite side of the boundary. These equilibrium distributions were obtained with the solid node velocity extrapolated from the flow field. Following this work, several researchers have introduced other boundary condition methods which also treat curved boundaries. Mei et al. [2] noted that Filippova and Hänel's method [1] is unstable when the boundary is located less than half a lattice space from the fluid node, and proposed to use the second neighboring fluid node to reduce the instability caused by the velocity extrapolation. Guo et al. [3] further improved the stability by assuming that the non equilibrium particle distributions on the solid node are identical to those of the fluid node. These are then used in a relaxation process, together with the equilibrium distributions obtained through extrapolation, to finally obtain the distributions to be streamed onto the fluid node. Variants of Guo et al.'s method have since been published [4,5] that basically modify the way the velocity is extrapolated on the solid node, but keep the non equilibrium distribution assumption.

Aside from the methods briefly described above, a scheme which does not rely on macroscopic fluid properties extrapolation has been introduced by Bouzidi et al. [6]. Their method interpolates the particle distribution at a location from which an emanating particle will be reflected by the boundary precisely on the fluid node, but in the opposite direction. The accuracy of this method has been improved by d'Humières and Ginzburg [7] with a boundary condition method they term multi-reflection, which uses three fluid node neighbors for the interpolation of particle distributions. Their intention is to increase the numerical accuracy from second to third order. One drawback of this method is the requirement for information at many node sites, which are not necessarily available for certain flow configurations, as noted by Chun and Ladd [8]. Also, as observed in Refs. [9,10], Bouzidi et al.'s method, which is based on distribution interpolation, does not conserve mass. We surmise that this also applies to higher order interpolation schemes, and constitutes a limitation of these methods.

Mass conserving methods have been proposed, for instance by Chen et al. [11], who tackled the problem from the standpoint of surface chemistry. They consider a boundary node which acts as a mass capacitor that accumulates and liberates mass into the fluid flow. They reported second order accuracy and total mass conservation for a Couette flow between concentric cylinders test case. The immersed boundary method, initially introduced by Peskin [12], is another type of mass conserving curved boundary method, which was adopted by Feng and Michaelides [13] to solve fluid-particle interaction problems in the lattice Boltzmann context. This method, which they call the immersed boundary lattice Boltzmann method (IB-LBM), assumes the presence of fluid inside an object and essentially relies on applying a body force to the fluid nodes near a boundary (inside and outside) which makes the fluid velocity match the boundary velocity. This method has attracted a great deal of interest in the LBM community and has since been applied by many authors [14-23]. However, it is, by definition, not applicable to boundaries that are not immersed; that is, when no nodes are available on the other side.

Given the qualities and limitations of the boundary condition methods we have described, the objective of this work is to propose a versatile boundary condition method which:

- is applicable to both immersed and non immersed boundaries;
- does not rely on any non equilibrium distribution assumption; and
- conserves mass better than methods based on distribution interpolation.

To achieve this goal, we have designed a scheme that relies only on equilibrium distributions emanating from ghost nodes specifically located on the lattice, in order to avoid the necessity of evaluating non equilibrium distributions.

2. Lattice Boltzmann Method

The LBM consists of simulating the statistical behavior of an ensemble of particles over a lattice with finite speeds. It stems from the discrete Boltzmann equation into which a collision model is introduced to account for particle interactions. As such, the dynamics of a system of particles is represented at a mesoscopic rather than a macroscopic level, as in continuum mechanics. The works of Succi [24] and Chen and Doolen [25] provide excellent insight into the origin and development of the method. The macroscopic properties of a fluid can be retrieved through weighted averages, or moments, of the particle distributions over all discrete lattice speeds. In this paper, we consider 2D flows and use the nine-speed lattice shown in Figure 1, commonly referred to as the d2q9 lattice. The discrete Boltzmann equation is given in Eq. (1), with the terms $f_i(\vec{x}, t)$ representing the distributions of particles moving in a particular direction, and $\Omega_i(\vec{x}, t)$ representing the collision operator.

$$\frac{\partial f_i(\vec{x}, t)}{\partial t} + \vec{c}_i \cdot \vec{\nabla} f_i(\vec{x}, t) = \Omega_i(\vec{x}, t) \quad (1)$$

Converting the partial derivatives into material derivatives and then replacing them with finite differences provides the basis for the usual two-step collide and stream process.

$$f_i(\vec{x} + \vec{c}_i \Delta t, t + \Delta t) = f_i(\vec{x}, t) + \Delta t \Omega_i(\vec{x}, t) \quad (2)$$

$$\text{collision: } f_i^*(\vec{x}, t) = f_i(\vec{x}, t) + \Delta t \Omega_i(\vec{x}, t) \quad (3)$$

$$\text{propagation: } f_i(\vec{x} + \vec{c}_i \Delta t, t + \Delta t) = f_i^*(\vec{x}, t) \quad (4)$$

The macroscopic fluid density and momentum are recovered with Eqs. (5) and (6). For simplicity, the term (\vec{x}, t) is omitted in the remainder of the paper.

$$\rho = \sum_i f_i \quad (5)$$

$$\rho \vec{u} = \sum_i \vec{c}_i f_i \quad (6)$$

Bhatnagar et al. [26] were the first to introduce a collision model which makes it possible to recover the incompressible Navier-Stokes equations, as demonstrated in Ref. [25], for instance, through a Chapman-Enskog expansion. The LBM using this collision model is often called LBGK, because its operator is linear:

$$\Omega_i = \frac{1}{\Delta t_R} (f_i^{(0)} - f_i). \quad (7)$$

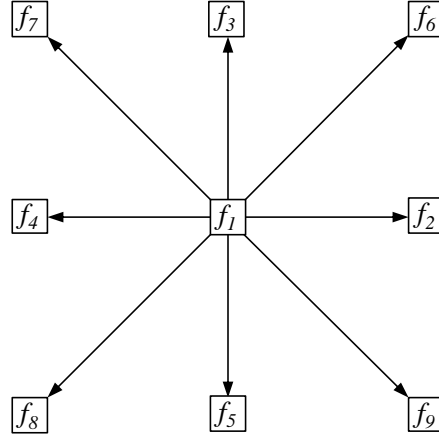


Figure 1: Schematic representation of the d2q9 lattice configuration.

The terms Δt_R and $f_i^{(0)}$ represent the relaxation time and equilibrium distribution functions respectively, and both relate to the macroscopic fluid properties. The relaxation time is a function of kinematic viscosity, and the equilibrium distributions are functions of the density and velocity. The parameters of the d2q9 LBGK are defined in Eqs. (8) to (12).

$$f_i^{(0)} = \rho w_i \left(1 + \frac{\vec{c}_i \cdot \vec{u}}{c_s^2} + \frac{1}{2} \frac{(\vec{c}_i \cdot \vec{u})^2}{c_s^4} - \frac{1}{2} \frac{\vec{u} \cdot \vec{u}}{c_s^2} \right) \quad (8)$$

$$\vec{w} = \left[\frac{4}{9} \quad \frac{1}{9} \quad \frac{1}{9} \quad \frac{1}{9} \quad \frac{1}{9} \quad \frac{1}{36} \quad \frac{1}{36} \quad \frac{1}{36} \quad \frac{1}{36} \right]^T \quad (9)$$

$$\vec{c}_i = c \begin{cases} (0,0), & \text{for } i=1 \\ \left(\cos\left((i-2)\frac{\pi}{2}\right), \sin\left((i-2)\frac{\pi}{2}\right) \right), & \text{for } i=2,\dots,5 \\ \left(\cos\left(\frac{\pi}{4} + (i-6)\frac{\pi}{2}\right), \sin\left(\frac{\pi}{4} + (i-6)\frac{\pi}{2}\right) \right), & \text{for } i=6,\dots,9 \end{cases} \quad (10)$$

$$c_s = \frac{c}{\sqrt{3}}, \quad c = \frac{\Delta x}{\Delta t} \quad (11)$$

$$p = \rho c_s^2, \quad \nu = c_s^2 \Delta t \left(\frac{\Delta t_R}{\Delta t} - \frac{1}{2} \right) \quad (12)$$

The LBGK is known to be unstable when the ratio $\Delta t/\Delta t_R$ approaches 2. In our study, we use the multiple relaxation collision operator, first introduced by d'Humières [27], instead of the LBGK. This method, commonly termed MRT, increases the numerical stability of the computations by relaxing moments of the distributions rather than the distributions themselves. Different relaxation times are applied to the moments, which separates the dynamic scales of the collision process. With the MRT, the collision step in Eq. (3) is replaced with:

$$|f\rangle^* = |f\rangle + \mathbf{M}^{-1} \mathbf{K} \mathbf{M} \left(|f\rangle^{(0)} - |f\rangle \right), \quad (13)$$

where \mathbf{K} is a diagonal matrix of relaxation coefficients, and \mathbf{M} is the matrix that shifts domain from the distribution space to the moment space. We have chosen the following orthogonal moment set, defined in Ref. [28]:

$$\begin{aligned}
 |m\rangle = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \\ m_7 \\ m_8 \\ m_9 \end{bmatrix} = \mathbf{M} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} = \begin{bmatrix} \langle 1 | \\ \langle c_x | \\ \langle c_y | \\ 3\langle c_x^2 + c_y^2 | - 4\langle 1 | \\ \langle c_x^2 - c_y^2 | \\ \langle c_x c_y | \\ -3\langle c_x^2 c_y | + 2\langle c_y | \\ -3\langle c_x c_y^2 | + 2\langle c_x | \\ 9\langle c_x^2 c_y^2 | - 6\langle c_x^2 + c_y^2 | + 4\langle 1 | \end{bmatrix} |f\rangle . \quad (14)
 \end{aligned}$$

Moments m_5 and m_6 correspond to the bulk and shear stress respectively, and their relaxation coefficients are set to $K_{5,5}=\Delta t/\Delta t_R$ and $K_{6,6}=\Delta t/\Delta t_R$. The other relaxation coefficients are typically set to 1. In Eqs. (13) and (14), the notations $| \rangle$ and $\langle |$ define the column and row vector expansions respectively.

3. Boundary Condition Description

The proposed boundary condition method uses an equilibrium distribution at a ghost node located at a distance $q = |\vec{c}_i| \Delta t_R$ from the fluid node that is missing a distribution after streaming, in the direction opposite to that of the missing distribution. The rationale behind this location selection is the linear nature of the BGK collision operator. If a collision process is carried out for a period $\Delta t = \Delta t_R$, a given distribution will be equal to its equilibrium value, and so its non equilibrium part will be null. Therefore, if we assume that the total collide and stream process time Δt is proportional to the distance between the sending and receiving nodes divided by the local lattice speed, and that the streaming process is performed instantly, the collision time alone becomes proportional to the inter-node distance. Our selection of q implies that the collision process time is Δt_R , which in turn makes it possible to eliminate the need to take into account the non equilibrium part of the distributions. We feel that this is a significant departure from methods proposed by other authors [3,5].

The fluid velocities used to compute the equilibrium distributions are interpolated or extrapolated from the pre-stream flow field and the velocities set on the boundary. The fluid densities can be computed with two methods:

1. Near wall approximation, which applies to straight or curved boundaries where the density is unknown on the other side.
2. Local interpolation, which is designed for the immersed version of our boundary condition scheme, when the density is also available on the other side.

Figure 2 shows an example of the velocity extrapolation points used for a straight boundary with tangential velocity, such as the top boundary in the lid driven cavity flow or the Couette flow between parallel plates. Figure 3 shows the velocity interpolation or extrapolation points used for a curved boundary, such as in the Couette flow between concentric cylinders or the cylinder in a free stream simulations. The simplest choice of interpolation or extrapolation points (a , b , and c) would consist of neighboring nodes along a given lattice direction; however, we have chosen to place these points along the boundary normal that intercepts the ghost node, g . This choice is applicable to both straight and curved boundaries, and was

made as a provision for the future implementation of slip velocity control through the adjustment of the tangential velocity.

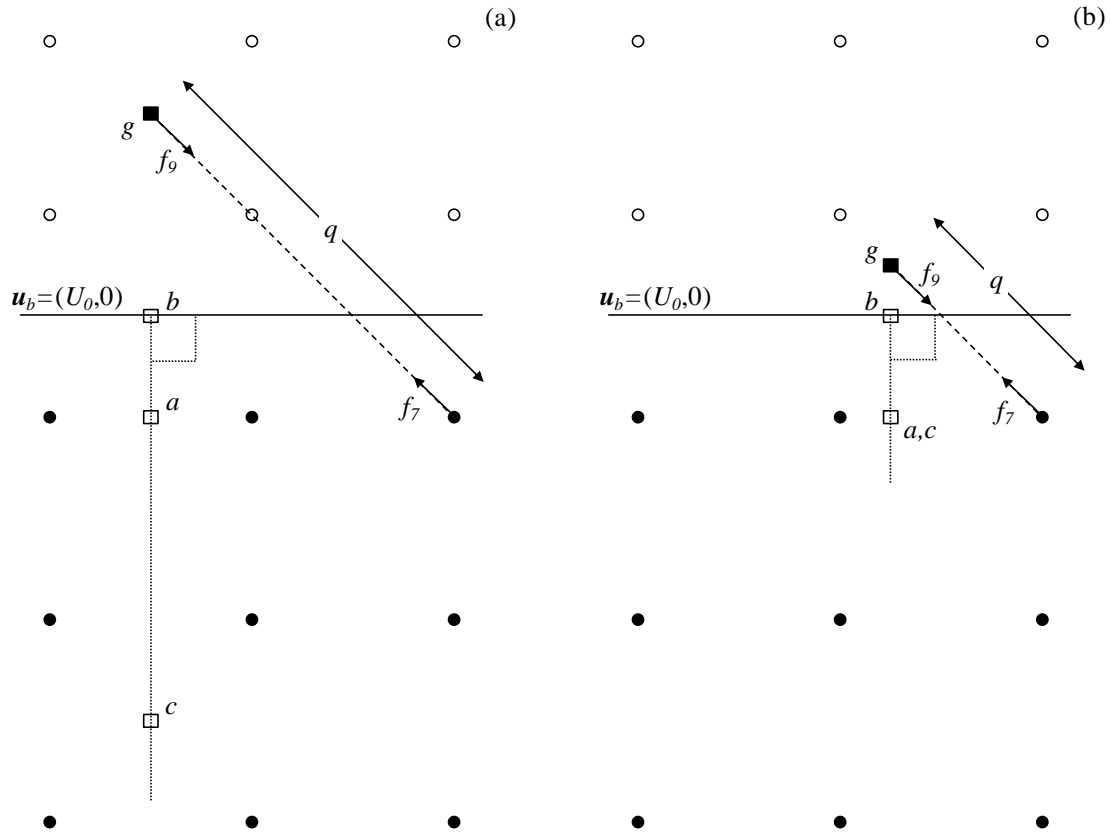


Figure 2: Extrapolation points for a straight boundary case. Filled circles are fluid nodes, open circles are nodes outside the domain or in a solid object, filled squares are the ghost nodes, and open squares are the points used in the extrapolation. The corresponding relaxation times are: a) $\Delta t_R = 1.5 \Delta t$, and b) $\Delta t_R = 0.75 \Delta t$.

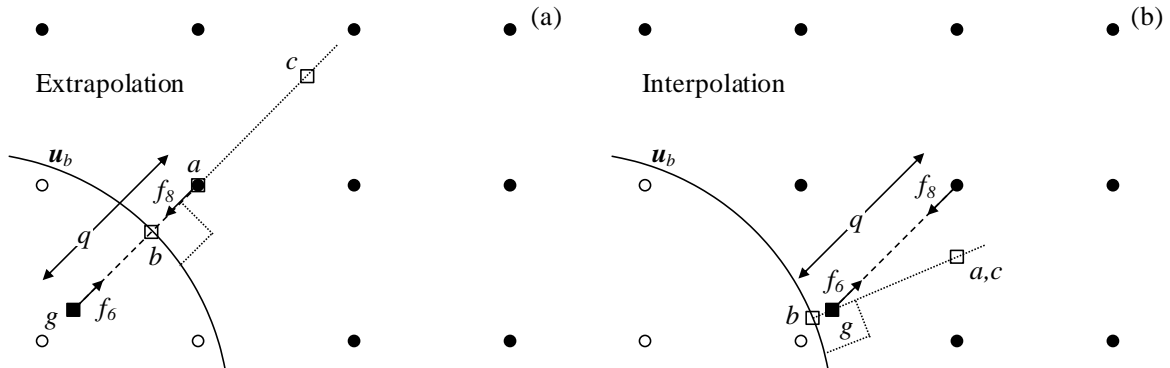


Figure 3: Extrapolation and interpolation points for a curved boundary case with $\Delta t_R = 0.8 \Delta t$. Filled circles are fluid nodes, open circles are nodes outside the domain or in a solid object, filled squares are the ghost nodes, and open squares are the points used in: a) extrapolation, or b) interpolation.

The velocity at a ghost node is obtained with the following linear set of equations:

$$\vec{u}_g = k\vec{u}_{g,1} + (1-k)\vec{u}_{g,2} , \quad (15)$$

$$\vec{u}_{g,1} = \vec{u}_a + k_1(\vec{u}_b - \vec{u}_a) , \quad (16)$$

$$\vec{u}_{g,2} = \vec{u}_c + k_2(\vec{u}_b - \vec{u}_c) , \quad (17)$$

$$k = \min \left(1, \frac{|\vec{x}_b - \vec{x}_a|}{|\vec{x}_g - \vec{x}_a|} \right) , k_1 = \frac{|\vec{x}_g - \vec{x}_a|}{|\vec{x}_b - \vec{x}_a|} , k_2 = \frac{|\vec{x}_g - \vec{x}_c|}{|\vec{x}_b - \vec{x}_c|} . \quad (18)$$

Eqs. (16) and (17) express linear interpolations or extrapolations at the ghost node relying on points a - b and c - b respectively. Eq. (15) expresses a linear combination introduced to reduce the instabilities that can be created when the coefficient k_i is too large, for instance when the boundary point b is very close to a . Point a is chosen such that it is as close as possible to the boundary and its bilinear interpolation relies only on fluid nodes. Point c is chosen such that its distance from the ghost node is large enough for the extrapolation coefficient k_2 to be smaller than or equal to 1.5. Therefore, if the boundary is very close to a , the linear combination in Eq. (15) puts more weight on Eq. (17) and point c , which increases the scheme's stability.

The fluid properties at points a and c are interpolated in the flow field using a bilinear interpolation. Since the interpolation or extrapolation at the ghost node is linear, we did not consider that it was necessary to use a higher order for the field interpolation. In Figure 2, the top boundary is placed exactly half a lattice space above the top row of fluid nodes, but its position could have been set arbitrarily, in which case the location of points a and c would be chosen so as to maintain the scheme's stability, as described above.

Finally, the equilibrium distributions on the ghost nodes are obtained with

$$f_i^{eq} = \rho_m w_i \left(1 + \frac{\vec{c}_i \cdot \vec{u}_g}{c_s^2} + 0.5 \frac{(\vec{c}_i \cdot \vec{u}_g)^2}{c_s^4} - 0.5 \frac{(\vec{u}_g \cdot \vec{u}_g)^2}{c_s^2} \right) , \quad (19)$$

where ρ_m is set according to the scheme selected:

1. Near wall approximation:

$\rho_m = \rho_a$ is the density interpolated bilinearly at point a , or

$\rho_m = \rho_0$ is the nominal fluid density.

In our experience, in order to conserve mass within the computational domain, it is better to choose $\rho_m = \rho_0$ for a boundary driving the velocity field, and $\rho_m = \rho_a$ for the other boundaries on which our method is applied.

2. Local interpolation for immersed boundaries:

$\rho_m = \rho_g$ is the density obtained through bilinear interpolation at the ghost node.

The proposed method somewhat resembles the work of Tiwari and Vanka [5], in the sense that in their study the velocity is also interpolated in the flow field at a point normal to the boundary. However, our method is fundamentally different. In Ref. [5], the field interpolation point location is set in order to obtain a numerically stable extrapolation of the velocity on the solid node. The authors also extrapolate non equilibrium distributions on that node. As we mentioned earlier, our method does not rely on non equilibrium distributions, and in some cases the ghost node velocity may come from an interpolation rather than an extrapolation. A velocity extrapolation approach similar to that in Ref. [5] has also been taken by Berthelsen and Faltinsen [29] for treating immersed boundaries with a Navier-Stokes solver, and we use some of their results as a basis for comparison with ours.

2.1 Immersed Treatment

As mentioned above, for the immersed version of our boundary condition method, the density ρ_g is obtained through local bilinear interpolation. The ghost node velocity determination is treated in the same manner for both sides of the boundary. For the inner, or solid, side of the boundary, points a and c are positioned using the reversed normal direction at the boundary. However, a major difference occurs after the ghost node distributions calculated with Eq. (19) are streamed. We apply a mass conservative velocity correction term to all the distributions of all solid nodes to ensure that their velocity is equal to the object velocity:

$$f_i = f_i + \frac{\Delta t}{c_s^2} \rho w_i (\vec{a}_F \cdot \vec{c}_i) , \quad (20)$$

where \vec{a}_F is an acceleration term computed with

$$\vec{a}_F = \frac{c_s^2}{\Delta t} \begin{bmatrix} \frac{1}{\sum_i w_i c_{i,x}^2} & 0 \\ 0 & \frac{1}{\sum_i w_i c_{i,y}^2} \end{bmatrix} \begin{bmatrix} u_{target,x} - u_x \\ u_{target,y} - u_y \end{bmatrix} , \quad (21)$$

and \vec{u}_{target} is the target velocity at a given node location. Eqs. (20) and (21) were derived from the common body force term used in the LBM [30],

$$\Delta f_i = \frac{\Delta t}{c_s^2} w_i (\vec{F} \cdot \vec{c}_i) , \quad (22)$$

that we used to construct a linear set of equations expressing the velocity variation as a function of the applied force \vec{F} :

$$\Delta u_x = \frac{1}{\rho} \sum_i \Delta f_i c_{i,x} = \frac{1}{\rho} \sum_i \frac{\Delta t}{c_s^2} w_i (\vec{F} \cdot \vec{c}_i) c_{i,x} , \quad (23)$$

$$\Delta u_y = \frac{1}{\rho} \sum_i \Delta f_i c_{i,y} = \frac{1}{\rho} \sum_i \frac{\Delta t}{c_s^2} w_i (\vec{F} \cdot \vec{c}_i) c_{i,y} , \quad (24)$$

$$\frac{\Delta t}{\rho c_s^2} \begin{bmatrix} \sum_i w_i c_{i,x}^2 & \sum_i w_i c_{i,x} c_{i,y} \\ \sum_i w_i c_{i,x} c_{i,y} & \sum_i w_i c_{i,y}^2 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} \Delta u_x \\ \Delta u_y \end{bmatrix} . \quad (25)$$

Note that the term $\sum_i w_i c_{i,x} c_{i,y}$ reduces to 0. Therefore, solving the system of Eq. (25) yields a decoupling

of forces and velocity variations, that is, F_x depends only on Δu_x , while F_y depends only on Δu_y . We have chosen to express Eqs. (20) and (21) in terms of acceleration rather than force, to better illustrate that our method represents a velocity correction.

3. Results

Four test cases were chosen to evaluate the proposed boundary condition method. The first test case is the well known lid driven cavity flow, which has been studied extensively and provides a good set of published numerical results, including the data of Ghia et al. [31]. The second and third test cases are the Couette flow between parallel plates and concentric cylinders, which were chosen because they provide analytical solutions that allow us to measure the error decay order for straight and curved boundaries. Finally, the cylinder in a free stream was selected for its available reference data, both numerical and

experimental, and because it is a test case well suited to demonstrating the immersed version of our boundary condition scheme.

We also compare our scheme to the interpolation bounce-back scheme of Bouzidi et al. [6] and the extrapolation scheme of Guo et al. [3]. The former was chosen because it does not rely on the extrapolation of macroscopic properties, and the latter because it uses an extrapolation of velocity combined with an assumption for non equilibrium distribution, which our method avoids. From here on, we'll simply refer to these methods as the Bouzidi method and the Guo method respectively.

3.1 Lid Driven Cavity Flow

For this test case, the domain is square with the outer layer of lattice nodes located half a lattice space from the boundary. On the bottom and side boundaries, the bounce-back scheme is applied. We selected a Reynolds number of 10^4 and compare our numerical results with those of Ghia et al. [31] and Erturk et al. [32], which were both produced by solving a vorticity stream function formulation of the Navier-Stokes equations. We used a lattice of 128×128 cell centered nodes, which is similar to the grid resolution in Ref. [31], where 129×129 edge nodes were used. Erturk et al. performed the computations with a finer grid resolution of 257×257 edge nodes. Also, to ensure that no mass leakage occurs, we set the ghost node densities to $\rho_m = \rho_0$. Figure 4 displays the mid-plane normalized horizontal and vertical velocities against the normalized axis dimensions, where

$$x^* = 2\frac{x}{L} - 1, y^* = 2\frac{y}{H} - 1, \quad (26)$$

and L and H represent the width and height of the domain respectively.

The results we obtained with the Guo extrapolation method and our method compare well with the selected reference data. The Bouzidi interpolation method produces the correct velocity profiles with slightly larger velocity amplitudes.

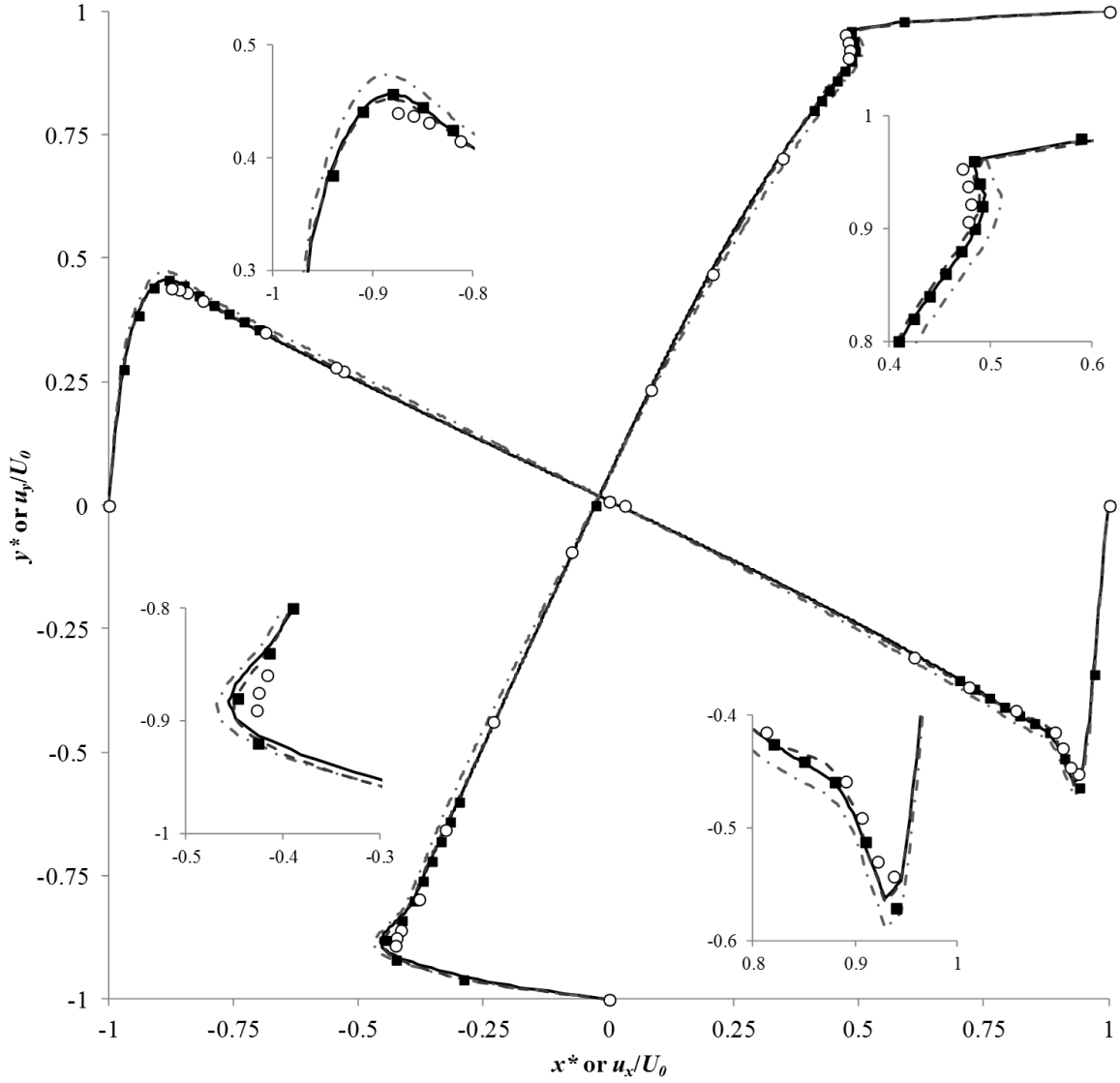


Figure 4: Mid-plane velocity profiles for the lid driven cavity flow at $Re=10^4$. White circles are from Ref. [31] and black squares are from Ref. [32]. The lines pertain to the boundary condition schemes of Bouzidi (dash-dotted), Guo (dashed), and ours (plain).

3.2 Couette Flow between Parallel Plates

For this test case, we use the same computational domain setup as for the lid driven cavity flow, but the left and right bounce-back boundaries are replaced by periodic boundaries. Simulations are conducted for a series of Reynolds number $Re=U_0 N_y/\nu$, chosen to produce a range of $\Delta t_R/\Delta t$ that allows us to study the effect of ghost node distance on the simulation error. This error is computed with the following L^2 -norm, using the analytical solution for the Couette flow:

$$error = \sqrt{\frac{1}{N_x N_y} \sum_i \left(\frac{u_{x,i} - u_{x,analytical}(y_i)}{U_0} \right)^2}, \quad (27)$$

$$u_{x,analytical}(y) = U_0 y/H, \quad (28)$$

where the i index of the summation covers all nodes. For each Reynolds number simulated, $N_y=N_x$ was varied from 4 to 80, and the nominal velocity U_0 was adjusted to maintain $\Delta t_R/\Delta t$ constant. The Mach number U_0/c_s never exceeded 0.1. Again, the ghost node densities were set to $\rho_m=\rho_0$.

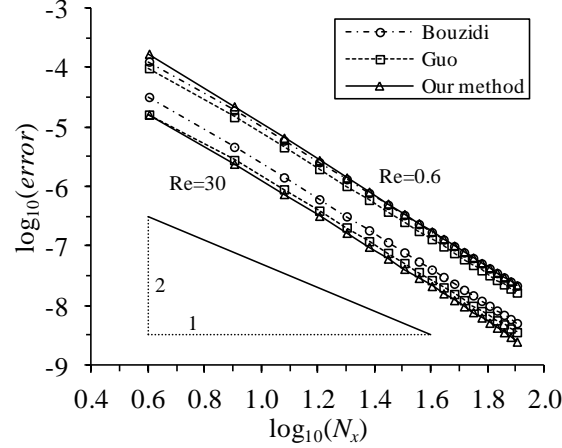


Figure 5: Simulation error for the Couette flow between parallel plates as a function of domain discretization for the smallest and largest Reynolds numbers.

Figure 5 displays the error evolution for the minimum and maximum Reynolds numbers simulated. For all Reynolds numbers, the error is reduced exponentially with increasing N_x , and the rate of error decay is always higher than 2. Figure 6 shows the error for the finest domain discretization, i.e. for $N_x=80$, as a function of $\Delta t_R/\Delta t$. We observe a nonlinear dependency, and all boundary condition methods produce an error pattern with a minimum located between $\Delta t_R/\Delta t=0.7$ and $\Delta t_R/\Delta t=0.85$. Our method and that of Guo, both based on extrapolation or interpolation of velocity, generate similar levels of error. However, in our method, the error grows faster when $\Delta t_R/\Delta t$ is larger than 1. Near $\Delta t_R/\Delta t=0.5$, our method's error is slightly smaller than that of Guo. These observations are in line with the way our method has been constructed. Since for this test case we placed the top boundary exactly half a lattice spacing above the top node row, when $\Delta t_R/\Delta t$ is close to 0.5, the ghost node is very close to the boundary, and its velocity, obtained with Eq. (15), also becomes very close to the actual boundary velocity, with the result that the error attributable to extrapolation is reduced. In contrast, when $\Delta t_R/\Delta t$ is larger than 1, our method extrapolates the velocity at a greater distance than the method of Guo, which reduces the relative accuracy.

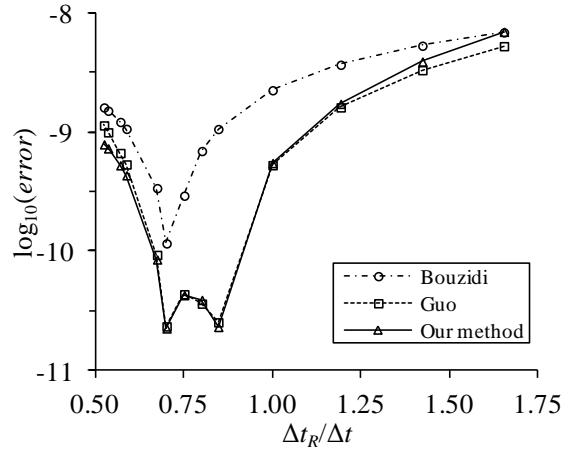


Figure 6: Simulation error for the Couette flow between parallel plates as a function of $\Delta t_R/\Delta t$, with N_x set to 80.

3.3 Couette Flow between Concentric Cylinders

For this simulation, we again use the square domain, and insert two concentric cylinders of radii r_2 and $r_1 = \beta r_2$ in the middle of the domain. The outer cylinder is fixed, and the inner cylinder rotates counter-clockwise with angular velocity ω_1 . The simulation is carried out over the entire domain, and curved boundary conditions are applied on both sides of each cylinder. Bounce-back is applied on all the square domain boundaries. For the near wall density approximation of our method, we set $\rho_m = \rho_0$ on the inner cylinder and $\rho_m = \rho_a$ on the outer cylinder, because the inner cylinder is the velocity driver. Ref. [3] provides the analytical solution for this problem, which exists for the tangential velocity between the cylinders, as follows:

$$u_{\theta,analytical}(r) = \frac{\omega_1 r_1 \beta}{1 - \beta^2} \left(\frac{r_2}{r} - \frac{r}{r_2} \right). \quad (29)$$

We compare our numerical results with the analytical solution with the following L^2 -norm:

$$error = \sqrt{\frac{1}{N_i} \sum_i \left(\frac{u_{\theta,i} - u_{\theta,analytical}(r_i)}{\omega_1 r_1} \right)^2}, \quad (30)$$

where N_i represents all the nodes located between the cylinders. We selected the same simulation parameters as given in Ref. [3], that is, $\beta = 0.5$, and $Re = \omega_1 r_1 (r_2 - r_1) / \nu = 5, 10, 20$, and 30. The domain is discretized so that the dimension r_1 is covered with 8 to 64 lattice nodes, i.e. $N_x = 40$ to 320. However, unlike the work described in Ref. [3], we did not keep $\Delta t_R / \Delta t$ constant across all Reynolds numbers. Instead, we elected to maintain the nominal Mach number at $N_x = 40$ constant, in order to remove the compressibility effects from our results comparison.

Figure 7 displays the error evolution as a function of domain discretization for all Reynolds numbers simulated, and our boundary condition method with near wall density approximation. The slopes of the linear curve fits are all smaller than -2, indicating at least a second order accuracy. The results presented in Table 1 indicate that all the boundary condition methods tested produce accuracies of at least second order for the selected simulation parameters. Furthermore, a comparison of the error at the finest domain discretization is shown in Figure 8. Overall, our method yields slightly smaller errors than either the Bouzidi or the Guo method.

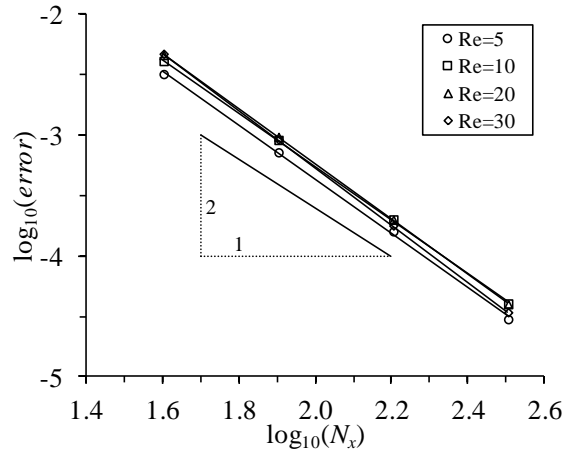


Figure 7: Simulation error for the Couette flow between concentric cylinders as a function of domain discretization. Our boundary condition method with near wall approximation for density is used.

		Bouzidi		Guo		Our method		Our method, immersed version	
Re	$\Delta t_R/\Delta t$	slope	error _{min}	slope	error _{min}	slope	error _{min}	slope	error _{min}
5	0.600	-2.12	-4.20	-2.16	-4.30	-2.22	-4.52	-2.22	-4.53
10	0.550	-2.12	-4.10	-2.16	-4.17	-2.20	-4.39	-2.20	-4.40
20	0.525	-2.13	-4.07	-2.23	-4.19	-2.26	-4.40	-2.25	-4.40
30	0.517	-2.14	-4.07	-2.32	-4.26	-2.34	-4.47	-2.33	-4.47

Table 1: Slopes of error decay and minimum errors for the Couette flow between concentric cylinders.

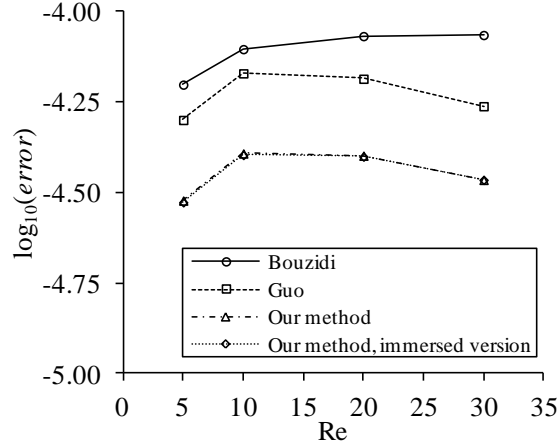


Figure 8: Minimum error as a function of Reynolds number for the Couette flow between concentric cylinders, with N_x set to 320.

Although all the boundary condition methods tested yield a numerical accuracy of at least second order, it is important to verify that they also conserve mass. It has been pointed out, in Ref. [10] for instance, that the Bouzidi method is not mass conserving. Our results corroborate this observation. Figure 9 presents the evolution of the mass between the cylinders for the case where $Re=30$ and $N_x=320$. Our method and that of Guo do conserve mass, but the Bouzidi method does not, nor does the immersed version of ours, for which the mass slowly increases. The rate of mass growth is, however, one order of magnitude smaller for the immersed version of our method than it is for the Bouzidi method.

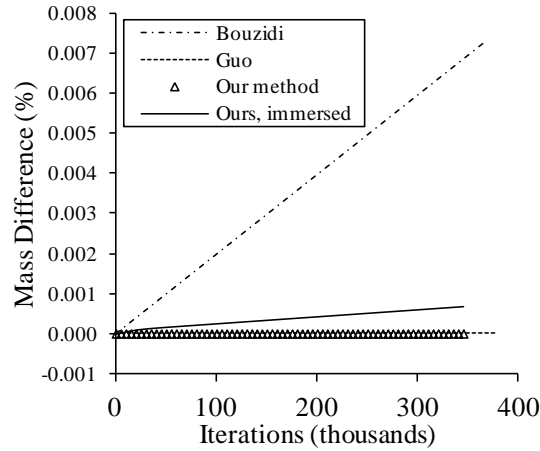


Figure 9: Variation of mass between the cylinders for $Re=30$ and $N_x=320$.

3.4 Cylinder in a free stream

We chose this test case specifically to demonstrate the ability of our immersed boundary condition method to produce a smooth density, or pressure, field across the boundary. The computational domain is square, with the cylinder located in the center and the fluid flowing from left to right.

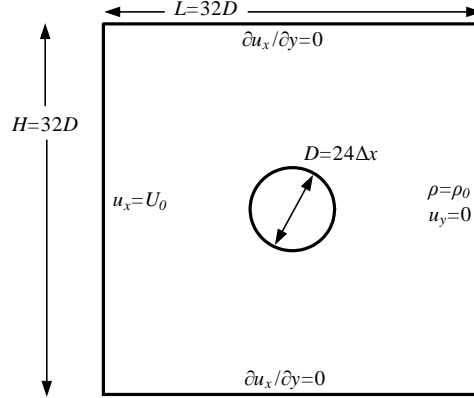


Figure 10: Domain geometry for the cylinder in a free stream.

During the streaming step, the distributions incoming from the domain exterior are unknown. We approximate the missing post-streaming distributions with those of the neighboring nodes located along the direction normal to the boundary. For example, a node located at the bottom of the domain is missing f_3, f_6 , and f_7 after streaming, and these distributions are approximated as $f_3(i,j)=f_3(i,j+1)$, $f_6(i,j)=f_6(i,j+1)$, and $f_7(i,j)=f_7(i,j+1)$. Then, the domain boundary conditions we impose modify all the boundary node distributions. This is accomplished in a two-step process. First, we use the velocity correction term of Eq. (20) to impose $u_x=U_0$ on the inlet, $u_y=0$ on the exit, and $\partial u_x / \partial y=0$ on the top and bottom boundaries. For the latter case, the x -component of velocity of the adjacent row nodes is used to fulfil our zero derivative condition, by correcting the boundary node u_x to make it equal to that of the adjacent row, that is, we let $u_x(i,j)=u_x(i,j-1)$ for the top boundary and $u_x(i,j)=u_x(i,j+1)$ for the bottom boundary. Second, we impose a constant pressure, or density, on the exit nodes in a way that aims to accelerate convergence by reducing pressure wave reflections in the domain:

$$|\Delta f\rangle_{exit} = \left(\rho_{exit}^* / \rho_{exit} - 1 \right) |f\rangle_{exit}, \quad (31)$$

$$\rho_{exit}^* = \rho_{exit} + \frac{c_s \Delta t}{2L} (\rho_0 - \rho_{exit}). \quad (32)$$

The term $c_s \Delta t / 2L$ is a relaxation factor that represents the ratio of the distance traveled by a sound wave over one time step to the distance it would need to travel to reach the inlet of the domain and be reflected back to the exit.

We performed a sensitivity analysis with the immersed version of our proposed boundary condition method and $Re_D=40$, and determined that the domain size and discretization illustrated in Figure 10 were sufficient for the drag coefficient to be quasi-independent of the grid dimensions. We calculated the aerodynamic coefficients for Reynolds numbers ranging from 20 to 200, using the momentum exchange method [33]. Table 2 and Table 3 present a comparison of our results for $Re=20, 40, 100$, and 200 with those reported by other authors ([11,22,29,34-39]). The subscripts m and a represent the mean value and amplitude.

Re	Source	$C_{d,m}$
20	Our method, immersed version	1.994
	Izham et al. [34]	1.998
	He et Doolen [35]	2.152
	Guo et Zhao [36]	2.048
	Lima E Silva et al. [37]	2.04
40	Our method, immersed version	1.493
	Izham et al. [34]	1.488
	He et Doolen [35]	1.499
	Guo et Zhao [36]	1.475
	Lima E Silva et al. [37]	1.54

Table 2: Aerodynamic coefficients for the cylinder in a free stream flow with $Re=20$ and $Re=40$.

Re	Source	$C_{d,m}$	$C_{d,a}$	$C_{l,a}$	St
100	Our method, immersed version	1.330	0.010	0.324	0.165
	Izham et al. [34]	1.308	0.023	0.342	0.166
	Arumuga et al. [38]	1.290		0.289	
	Berthelsen [29]	1.38	0.010	0.34	0.169
	Calhoun [39]	1.33	0.014	0.30	0.175
200	Our method, immersed version	1.330	0.046	0.668	0.197
	Izham et al. [34]	1.263	0.021	0.546	0.195
	Chen et al. [11]	1.329			0.195
	Berthelsen [29]	1.37	0.046	0.70	0.200
	Calhoun [39]	1.172	0.058	0.67	0.202
	Nishida et Meichin [22]	1.346	0.0201	0.6557	0.202

Table 3: Aerodynamic coefficients for the cylinder in a free stream flow with $Re=100$ and $Re=200$.

Our results are in good agreement with those reported by other authors, including the works of Izham et al. [34], Arumuga et al. [38], and Chen et al. [11], who also use the LBM on a regular lattice. However, the square computational domain of Ref. [34] was smaller and coarser than ours, with a width spanning 16 cylinder diameters and 256 lattice spacings. Unfortunately, the applied domain boundary conditions were not specified in Ref. [34], and we could not reproduce domain independent results using their domain size. He and Doolen [35] and Guo and Zhao [36] also used the LBM, but on curvilinear coordinates, with a finer lattice resolution near the cylinder. Comparison of results, especially for $Re=40$, demonstrates that our curved boundary condition method produces similar drag coefficients without the need to conform the lattice to the cylinder geometry. Nishida and Meichin [22] used the LBM with an immersed method they called Seamless Immersed Boundary, because it produces smooth pressure fields across the boundary. We make the same claim for our immersed method, which we demonstrate below.

Figure 11 presents the Strouhal number as a function of Reynolds number. The transition from steady to unsteady flow occurs between $Re=46$ and $Re=46.5$. This is in excellent agreement with the critical Reynolds number of 46 cited by Higuera and Succi [40].

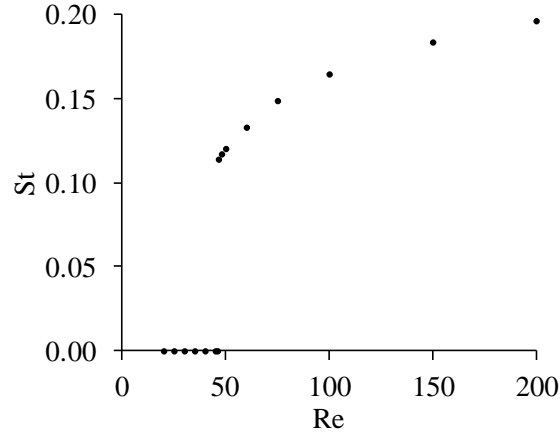


Figure 11: Strouhal number for the cylinder in a free stream.

Figure 12 shows the normalized density field over the cylinder for $Re=100$ at an arbitrarily selected time. The contour lines were calculated with linear interpolations and no further smoothing was applied. The density, or pressure, varies smoothly across the boundary. The velocity is null everywhere inside the cylinder, but the pressure is not uniform because of the velocity correction term of Eq. (20) that we apply. This is actually the key to ensuring the smooth pressure transition. Furthermore, we believe that this will be an advantage if one wants to study moving boundaries with our method. The pressure field continuity will reduce the local pressure waves induced when a node switches status from solid to fluid, or vice-versa.

The pressure field smoothness is further demonstrated through the plot of pressure coefficient against angle from leading edge, presented in Figure 13. We tested all boundary condition methods at $Re=40$, for which reference experimental data is available from Grove et al. [41]. Numerical data computed with a finite element method are also available from Pontaza et al. [42]. All boundary condition methods capture the pressure coefficient profile well. This profile was calculated in the same way for all methods, using an extrapolation from the flow field outside the cylinder. However, the methods of Bouzidi and of Guo introduce more waviness near the angle ranges of 10° to 30° , and 60° to 80° . For the numerical domain we used, these angle ranges correspond to locations where the cylinder surface normals are the least aligned with any of the lattice directions. We surmise that our method produces less wavy pressure coefficient profiles because it uses velocity interpolation points located normal to the boundary, whereas the methods of Bouzidi and of Guo use neighboring node locations along the lattice directions.

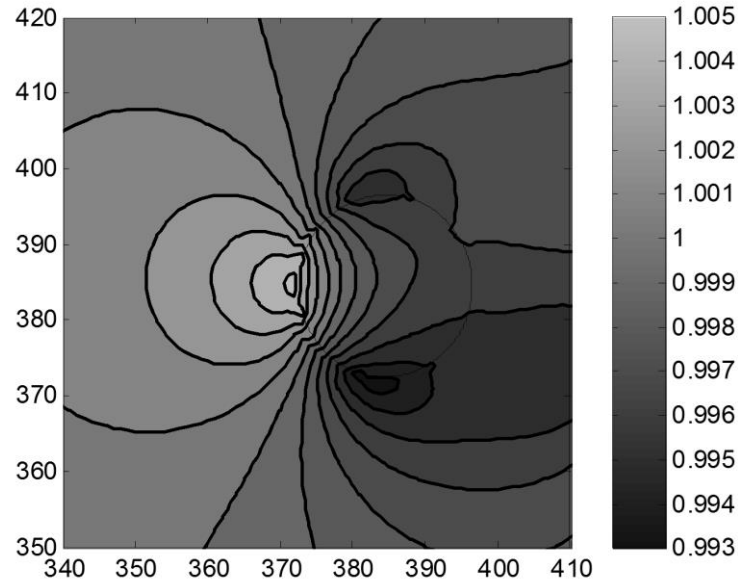


Figure 12: Density normalized with ρ_0 over the cylinder, for $Re=100$ at an arbitrarily selected time. Contour lines are in steps of 0.001 from 0.993 to 1.005.

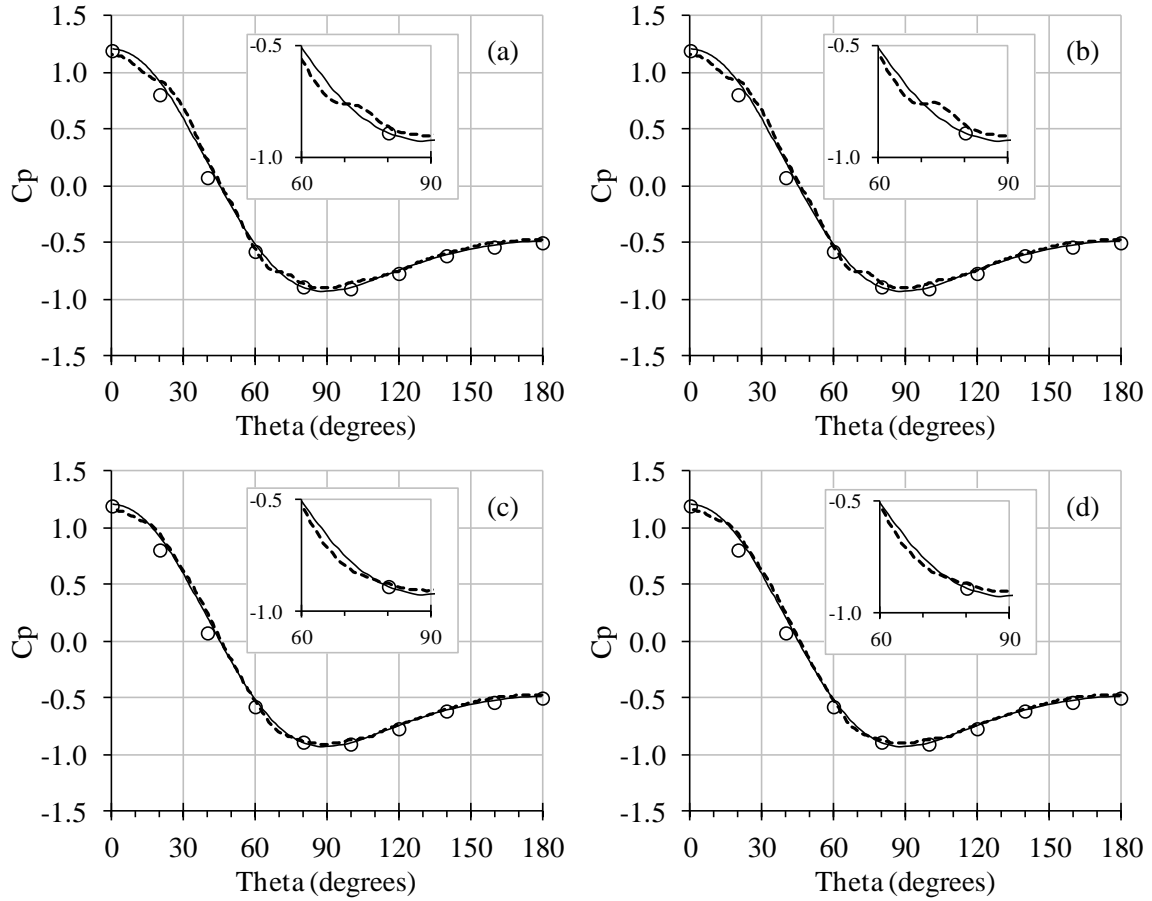


Figure 13: Pressure coefficient on the cylinder surface at $Re=40$. Open circles represent the experimental results of Grove et al. [41]. Plain lines represent the numerical results of Pontaza et al. [42]. Dashed lines represent the boundary condition methods tested: a) Bouzidi, b) Guo, c) ours, and d) the immersed version of ours.

4. Discussion

The proposed boundary condition was tested on four standard cases. For the lid driven cavity flow, the velocity fields obtained for $Re=10^4$ agreed well with the numerical results in Refs. [31,32], which were computed by solving a vorticity stream function formulation of the Navier-Stokes equations. For the Couette flow between parallel plates or concentric cylinders, the solution accuracy was at least of second order. For the cylinder in a free stream simulation with the immersed version of our boundary condition method, the aerodynamic coefficients obtained for Re ranging between 40 and 200 agreed well with values published by other authors, including Refs. [35,36], in which the LBM on curvilinear coordinates was used. Our boundary condition method also yielded the correct Re for the onset of vortex shedding, and produced smooth density contours across the cylinder boundary.

We compared our method to the interpolation bounce-back scheme of Bouzidi and the velocity extrapolation scheme of Guo. For the Couette flow test cases, the Bouzidi method produced larger errors than either the Guo method or our method, and, with $\Delta t_R/\Delta t$ close to 0.5, our method yielded smaller errors than the Guo method. However, as $\Delta t_R/\Delta t$ increased beyond 1, our method became less accurate than Guo's. This is a logical consequence of how our method was constructed. With $\Delta t_R/\Delta t$ larger than 1, the ghost node we use to extrapolate velocity is located beyond one lattice spacing, and so the error due to extrapolation is larger than it is for the Guo method. However, when the ghost node is located near a half lattice spacing, or when $\Delta t_R/\Delta t$ is near 0.5, the extrapolation distance is smaller. For a curved boundary, over a cylinder for instance, the velocity can actually be obtained with a combination of extrapolations and interpolations, as shown in Figure 3. We hypothesize that the interpolations increase the accuracy relative to schemes that rely only on extrapolations. Furthermore, for the cylinder in a free stream test case, both versions of our method produced surface pressure coefficients that are in better agreement with the selected numerical reference data than the other two tested boundary condition methods.

5. Conclusion

We have introduced a versatile velocity boundary condition method for the LBM. The method we propose can be used in conjunction with straight or curved boundaries, whether immersed or not. It is based on the assumption that the collision process time is proportional to the inter node distance divided by the lattice speed. This assumption allows us to consider only the streaming of equilibrium distributions emanating from a ghost node, located relative to the fluid node in such a way as to make its collision time equal to the relaxation time Δt_R .

Our method appears to be well suited for flow conditions with low viscosity or high speeds, since $\Delta t_R/\Delta t$ approaches 0.5 in these conditions. The next step would be to test it on high Re flows, in conjunction with the cascaded collision operator [43] that is stable in the limit of zero viscosity.

The continuous density contours that we obtained open the way to using the immersed version of our method for simulating moving bodies on the lattice. The density, or pressure, continuity would alleviate the expected effects of local pressure waves caused by nodes changing status from inside to outside the object. However, fast algorithms would need to be developed to detect object-lattice interceptions and create all the utility data required to apply the boundary conditions, because these calculations will have to be repeated at every simulation time step.

Finally, the proposed method is better able to capture pressure coefficient profiles in regions where an object surface normal is not aligned with lattice directions. This may be an advantage, for instance when computing flows around an airfoil, because the surface curvature and pressure vary very quickly near the leading edge.

References

- [1] Filippova O, Hänel D. Lattice-Boltzmann simulation of gas-particle flow in filters. *Comput Fluids* 1997;26:697–712.
- [2] Mei R, Luo L-S, Shyy W. An accurate curved boundary treatment in the lattice Boltzmann method. *J Comput Phys* 1999;155:307–30.
- [3] Guo Z, Zheng C, Shi B. An extrapolation method for boundary conditions in lattice Boltzmann method. *Phys Fluids* 2002;14:2007.
- [4] Kang J, Kang S, Suh YK. A dynamic boundary model for implementation of boundary conditions in lattice-Boltzmann method. *J Mech Sci Technol* 2008;22:1192–201.
- [5] Tiwari A, Vanka SP. A ghost fluid Lattice Boltzmann method for complex geometries. *Int J Numer Methods Fluids* 2011.
- [6] Bouzidi M, Firdaouss M, Lallemand P. Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Phys Fluids* 2012;13:3452–9.
- [7] Ginzburg I, d’Humières D. Multireflection boundary conditions for lattice Boltzmann models. *Phys Rev E* 2003;68:66614.
- [8] Chun B, Ladd a. Interpolated boundary condition for lattice Boltzmann simulations of flows in narrow gaps. *Phys Rev E* 2007;75:066705.
- [9] Lallemand P, Luo L-S. Lattice Boltzmann method for moving boundaries. *J Comput Phys* 2003;184:406–21.
- [10] Rohde M, Kandhai D, Derksen J, Van den Akker H. Improved bounce-back methods for no-slip walls in lattice-Boltzmann schemes: Theory and simulations. *Phys Rev E* 2003;67:066703.
- [11] Chen S, Bao S, Liu Z, Li J, Yi C, Zheng C. A heuristic curved-boundary treatment in lattice Boltzmann method. *Europhys Lett* 2010;92:54003.
- [12] Peskin CS. The immersed boundary method. *Acta Numer* 2002;11:479–517.
- [13] Feng Z-G, Michaelides EE. The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems. *J Comput Phys* 2004;195:602–28.
- [14] Zhang J, Johnson PC, Popel AS. An immersed boundary lattice Boltzmann approach to simulate deformable liquid capsules and its application to microscopic blood flows. *Phys Biol* 2007;4:285–95.
- [15] Strack OE, Cook BK. Three-dimensional immersed boundary conditions for moving solids in the lattice-Boltzmann method. *Int J Numer Methods Fluids* 2007;55:103–25.
- [16] Shu C, Liu N, Chew Y, Lu Z. Numerical simulation of fish motion by using lattice Boltzmann-immersed boundary velocity correction method. *J Mech Sci Technol* 2007;21:1139–49.

- [17] Shu C, Wu J. A new immersed boundary-lattice Boltzmann method and its application to incompressible flows. *Mod Phys Lett B* 2009;23:261–4.
- [18] Pantula S, Lu M, Liou W. Calculations of turbulent flow around airfoils with attached flexible fin using an immersed boundary method. In: *Proceedings 47th AIAA Aerospace Sciences Meeting*; 2009.
- [19] Wu J, Shu C, Zhang YH. Simulation of incompressible viscous flows around moving objects by a variant of immersed boundary-lattice Boltzmann method. *Int J Numer Methods Fluids* 2010;62:327–54.
- [20] Liu T, Liu G, Ge Y, Wu H, Wu W. Extended lattice Boltzmann equation for simulation of flows around bluff bodies in high Reynolds number. In: *Proceedings BBAA VI International Colloquium on: Bluff Bodies Aerodynamics & Applications*; 2008.
- [21] Rojas R, Seta T, Hayashi K, Tomiyama A. Immersed boundary-finite difference lattice Boltzmann method using two relaxation times. In: *Proceedings Ninth International Conference on CFD in the Minerals and Process Industries*; 2012.
- [22] Nishida H, Meichin Y. Seamless immersed boundary lattice Boltzmann method for incompressible flow simulation. In: *Proceedings Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*; 2012.
- [23] Fu SC, So RMC, Leung WWF. A lattice Boltzmann and immersed boundary scheme for model blood flow in constricted pipes: Part 2 - pulsatile flow. *Commun Comput Phys* 2013;14:153–73.
- [24] Succi S. *The lattice Boltzmann equation for fluid dynamics and beyond*. Oxford University Press; 2001.
- [25] Chen S, Doolen GD. Lattice Boltzmann method for fluid flows. *Annu Rev Fluid Mech* 1998;30:329–64.
- [26] Bhatnagar PL, Gross EP, Krook M. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Phys Rev* 1954;94:511–25.
- [27] d’Humières D. Generalized lattice-Boltzmann equations. *Rarefied gas dynamics: Theory and simulations*, American Institute of Aeronautics and Astronautics, Inc.; 1994, p. 450–8.
- [28] Premnath K, Banerjee S. Incorporating forcing terms in cascaded lattice Boltzmann approach by method of central moments. *Phys Rev E* 2009;80:036702.
- [29] Berthelsen PA, Faltinsen OM. A local directional ghost cell approach for incompressible viscous flow problems with irregular boundaries. *J Comput Phys* 2008;227:4354–97.
- [30] Nguyen PH. Numerical simulations of the lattice Boltzmann method for determination of hydrodynamic properties of fractal aggregates. *Université de Genève*, 2007.
- [31] Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible using the Navier-Stokes equations multigrid method. *J Comput Phys* 1982;48:387–411.

- [32] Erturk E, Corke TC, Gökçöl C. Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers. *Int J Numer Methods Fluids* 2005;48:747–74.
- [33] Mei R, Yu D, Shyy W, Luo L-S. Force evaluation in the lattice Boltzmann method involving curved geometry. *Phys Rev E* 2002;65:041203.
- [34] Izham M, Fukui T, Morinishi K. Application of regularized lattice Boltzmann method for incompressible flow simulation at high Reynolds number and flow with curved boundary. *J Fluid Sci Technol* 2011;6:812–22.
- [35] He X, Doolen G. Lattice Boltzmann method on curvilinear coordinates system: Flow around a circular cylinder. *J Comput Phys* 1997;134:306–15.
- [36] Guo Z, Zhao T. Explicit finite-difference lattice Boltzmann method for curvilinear coordinates. *Phys Rev E* 2003;67:066709.
- [37] A.L.F. Lima E Silva, Silveira-Neto A, Damasceno JJR. Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. *J Comput Phys* 2003;189:351–70.
- [38] Arumuga PD, Kumar GVS, Dass AK. Lattice Boltzmann simulation of flow over a circular cylinder at moderate Reynolds numbers. *Therm Sci* 2012:93.
- [39] Calhoun D. A cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions. *J Comput Phys* 2002;176:231–75.
- [40] Higuera FJ, Succi S. Simulating the flow around a circular cylinder with a lattice Boltzmann equation. *Europhys Lett* 1989;8:517.
- [41] Grove AS, Shair FH, Petersen EE. An experimental investigation of the steady separated flow past a circular cylinder. *J Fluid Mech* 1964;19:60–80.
- [42] Pontaza JP, Reddy JN. Spectral/hp least-squares finite element formulation for the Navier–Stokes equations. *J Comput Phys* 2003;190:523–49.
- [43] Geier M, Greiner A, Korvink JG. Cascaded digital lattice Boltzmann automata for high Reynolds number flow. *Phys Rev E* 2006;73:066705.

ANNEXE C – ARTICLE 2 : AN IMPLEMENTATION OF THE SPALART-ALLMARAS TURBULENCE MODEL IN A MULTI-DOMAIN LATTICE BOLTZMANN METHOD FOR SOLVING TURBULENT AIRFOIL FLOWS

Cet article a été publié dans une revue scientifique avec comité de lecture. La version publiée diffère de la version soumise suite à des modifications effectuées par l'éditeur.

Pellerin N, Leclaire S, Reggio M. An implementation of the Spalart-Allmaras turbulence model in a multi-domain lattice Boltzmann method for solving turbulent airfoil flows. *Comput Math with Appl* 2015;70:3001-18

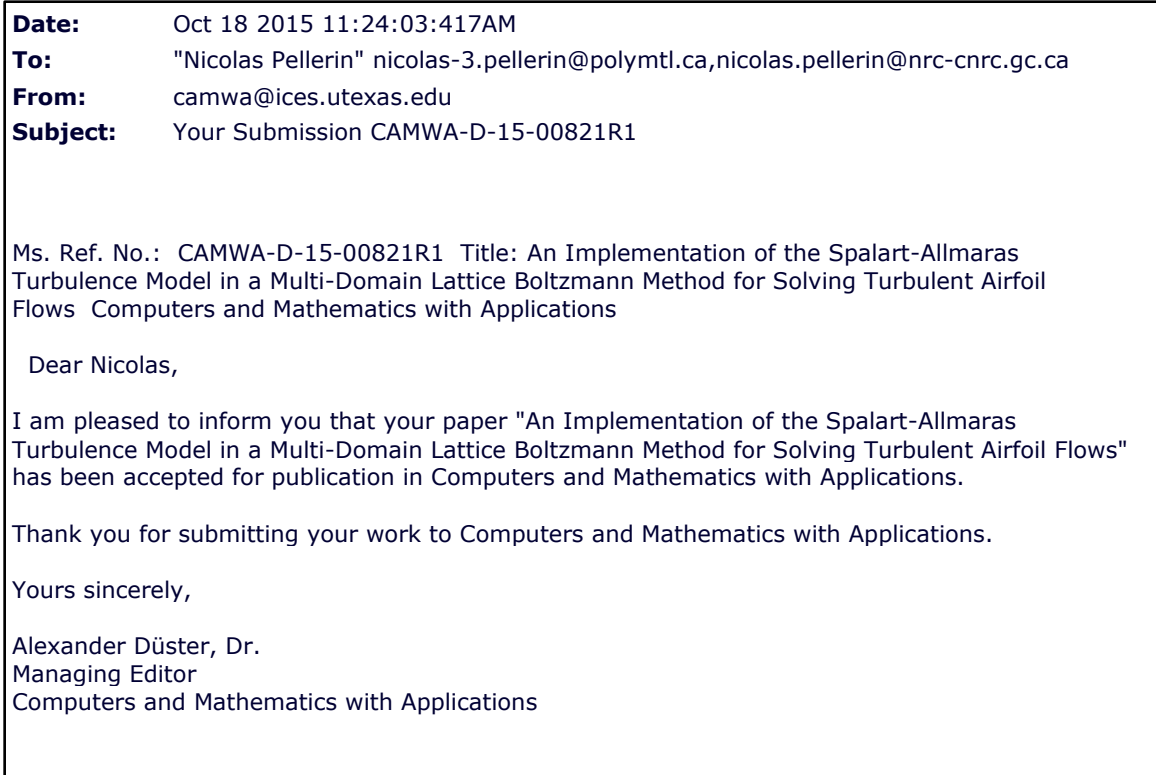


Figure C.1 : Avis d'acceptation par les éditeurs de l'article «An implementation of the Spalart-Allmaras turbulence model in a multi-domain lattice Boltzmann method for solving turbulent airfoil flows».

Title

An Implementation of the Spalart-Allmaras Turbulence Model in a Multi-Domain Lattice Boltzmann Method for Solving Turbulent Airfoil Flows

Author (corresponding)

Nicolas Pellerin

Department of Mechanical Engineering, Polytechnique Montréal, 2500 chemin de Polytechnique, Montréal, Québec H3T 1J4, Canada

nicolas-3.pellerin@polymtl.ca

(613) 297-6176

Co-authors

Sébastien Leclaire

Department of Mechanical Engineering, Polytechnique Montréal, 2500 chemin de Polytechnique, Montréal, Québec H3T 1J4, Canada

sebastien.leclaire@polymtl.ca

(514) 340-5121, ext. 5895

Marcelo Reggio

Department of Mechanical Engineering, Polytechnique Montréal 2500 chemin de Polytechnique, Montréal, Québec H3T 1J4, Canada

marcelo.reggio@polymtl.ca

(514) 340-5121, ext. 4411

Abstract

A methodology for solving turbulent airfoil flows based on the lattice Boltzmann method is proposed. It employs a multi-domain grid refinement approach, the cascaded collision operator, and a finite-difference implementation of the Spalart-Allmaras turbulence model. It is validated for the flow over a NACA0012 airfoil at a Reynolds number of 5×10^5 , and over the low Reynolds S1223 and E387 airfoils, both at $Re = 2 \times 10^5$. The results for the NACA0012 airfoil, in terms of force coefficients, pressure coefficients, and velocity profiles, compare favorably with the numerical results of two other studies, both of which use the Spalart-Allmaras turbulence model. The results for the other two airfoils successfully capture the experimental lift and drag profiles reported in the literature. Overall, the proposed methodology is shown to be appropriate for solving turbulent airfoil flows, provided the grid is sufficiently refined near the walls.

Keywords

Lattice Boltzmann method, boundary condition, turbulence, Spalart-Allmaras, airfoil.

Classification code

76G25

1. Introduction

The lattice Boltzmann method has become a popular computational tool for solving a variety of fluid flows, from low Reynolds number flows around bluff bodies [1,2] to immiscible multiphase flows [3,4]. Efforts have also been made to solve higher Reynolds number turbulent flows, around airfoils, for instance, by supplementing the LBM with turbulence models. Filippova et al. [5] coupled a multi-domain LBM solver to a finite-difference scheme of a two-equation $k-\varepsilon$ turbulence model, and they introduced the use of wall functions in the curved boundary condition formulation of Filippova and Hänel [6] to compute the flow around a NACA4412 airfoil. Imamura et al. [7] used a generalized coordinate formulation of the LBM to obtain a body-fitted refined mesh around a NACA0012 airfoil, which they coupled to the Baldwin-Lomax turbulence model to calculate drag and lift at a Reynolds number of 5×10^5 and an angle of attack of 7° . They claim that the force coefficients they obtained are almost identical to those produced with CFL3D [8], however the differences are non-negligible, particularly for lift. Li et al. [9] obtained flow solutions for an iced business jet airfoil using a multi-domain grid refinement setup, and their results compare well with experimental data. They modeled turbulence through a two equation renormalization group solved with a finite-difference scheme, and used a shear stress model near the walls in conjunction with the boundary condition formulation of Chen et al. [10] based on momentum control. The Spalart-Allmaras turbulence model has also been used, by Li et al. [11], in conjunction with a non-uniform Cartesian mesh refinement strategy. They obtained NACA0012 flow solutions at $Re=5 \times 10^5$ that agree reasonably well with numerical results produced with CFL3D [8], however their computational setup requires more than 4 million lattice sites in order to obtain a sufficiently fine resolution near the airfoil. Also, it can be noticed that, although they cited the original Spalart-Allmaras reference [12], Li et al. used the -fv3 modification of the model, which is no longer recommended because it has an odd effect on transition behavior at low Reynolds numbers [13].

Although these studies provide adequate strategies for solving turbulent airfoil flows, there are currently some drawbacks, mainly with the mesh employed. For instance, the body-fitted mesh of Imamura et al. [7] is well suited for a smooth airfoil, but it would be cumbersome to generate for an airfoil covered with arbitrary ice shapes. The non-uniform mesh of Li et al. [11] lacks versatility, because the mesh refinement cannot be purely local, that is, the distance between the neighboring rows or columns of lattice sites is the same over the entire domain. This makes it difficult to refine the mesh where it is mostly required, which is mainly in the boundary layer area. The approach of Li et al. [9] appears to be more appropriate, since the grid is progressively refined as the distance to the airfoil decreases, making it more easily applicable to arbitrary geometrical shapes. However, like Filippova et al. [5], they chose to use a two equation turbulence model, which adds complexity to the solver, as opposed to the algebraic Baldwin-Lomax model used by Imamura et al. [7] and the one equation Spalart-Allmaras model chosen by Li et al. [11].

To solve wall bounded turbulent flows, a very fine mesh is needed near the walls in order to solve the turbulent boundary layer appropriately. Typically, the y^+ of the first lattice site should be near 1. This requirement is often impractical, however, as the total number of sites can grow beyond computational capabilities. Filippova et al. [5] and Li et al. [9] addressed this drawback with the use of wall functions that model the underlying boundary layer and remove the fine mesh requirement. Such a model was not used by either Li et al. [11] or by Imamura et al. [7].

In light of the advantages and drawbacks identified in the above studies, we elected to build an LBM turbulent airfoil flow solver that uses a multi-domain configuration and the Spalart-Allmaras turbulence model. This approach makes it possible to refine a lattice over arbitrary shapes and to solve only one additional equation for modeling turbulence. We did not implement wall functions, because we are focusing on validating the combination of the S-A model and the multi-domain configuration.

Although resolving the additional equation departs from the inherent simplicity of the LBM, this step is required for treating of two-dimensional flows. All the turbulence scales must be modeled, since turbulence is a three-dimensional phenomenon. With our approach, the LBM unsteady flow solver coupled with the S-A model is run until a steady flow is obtained, a process that is analogous to solving the Reynolds-averaged Navier-Stokes equations. In order to avoid the need to resolve a turbulence transport equation, the Large Eddy Simulation (LES) technique could be used in conjunction with a turbulent viscosity based on a strain rate tensor computed locally with particle distributions. However, this technique should be applied in three dimensions. In fact, LES models are currently employed in commercial 3D LBM packages [14,15].

The proposed LBM flow solving strategy is validated for a NACA0012 airfoil at $Re=5 \times 10^5$, and the computational results are compared to those of Lockard et al. [8] obtained with CFL3D. We compare lift and drag coefficients, pressure coefficients, and velocity profiles for angles of attack at which the flow is still attached. We also compare our velocity profiles to those obtained by Li et al. [11]. The proposed LBM flow solver is also tested on the S1223 airfoil, which is designed to produce high lift at low Reynolds numbers, and the E387 airfoil. We compare lift and drag coefficients for $Re=2 \times 10^5$ to those obtained experimentally by Selig and Guglielmo [16] and McGhee et al. [17]. We chose this second validation case because the S1223 and E387 airfoils are not symmetrical, unlike the NACA0012 airfoil, and the relatively low Re makes it possible to limit the grid refinement requirement in the boundary layer.

This article is organized as follows. In section 2, we detail the LBM and the collision operator we have chosen. In section 3, we provide a description of the multi-domain algorithm based on a recursive construction. The implementation of the Spalart-Allmaras turbulence model is given in section 4, and the airfoil and domain boundary condition equations are provided in sections 5 and 6 respectively. The computational results obtained are provided in sections 7 and 8, and discussed in section 9.

2. Lattice Boltzmann Method

The LBM model we have chosen for solving high Reynolds number airfoil flows is the d2q9 lattice with the cascaded collision operator proposed by Geier et al. [18]. This operator was selected for its stability at low viscosity, or high Reynolds number [18]. Our implementation of the operator is identical to the procedure described by Leclaire et al. [19]. We present the details of the method using the compact Dirac $\langle \text{bra} |$ and $| \text{ket} \rangle$ notation for row and column vector expansions along the i -index of the finite lattice speeds. The particle distribution collision, expressed as

$$|f\rangle^* = |f\rangle + \mathbf{M}^{-1} |\Delta m\rangle, \quad (1)$$

is performed in the moment space of the distributions, and the moment variations are determined through the following cascade of operations:

$$\Delta m_1 = 0, \quad (2)$$

$$\Delta m_2 = 0, \quad (3)$$

$$\Delta m_3 = 0, \quad (4)$$

$$\Delta m_4 = \frac{\Delta t}{\Delta t_{R_4}} \left(2c^2 \rho + 3\rho(u_x^2 + u_y^2) - 3a_1 \right), \quad (5)$$

$$\Delta m_5 = \frac{\Delta t}{\Delta t_{R_5}} \left(\rho(u_x^2 + u_y^2) + a_2 \right), \quad (6)$$

$$\Delta m_6 = \frac{\Delta t}{\Delta t_{R_6}} \left(\rho u_x u_y - a_3 \right), \quad (7)$$

$$\Delta m_7 = \frac{\Delta t}{\Delta t_{R_7}} \left(6\rho u_x^2 u_y - 6u_x a_3 - 3u_y a_4 + 3a_5 \right) - 0.5u_y (\Delta m_4 + 3\Delta m_5) - 6u_x \Delta m_6, \quad (8)$$

$$\Delta m_8 = \frac{\Delta t}{\Delta t_{R_8}} \left(6\rho u_x u_y^2 - 6u_y a_3 - 3u_x a_6 + 3a_7 \right) - 0.5u_x (\Delta m_4 - 3\Delta m_5) - 6u_y \Delta m_6, \quad (9)$$

$$\Delta m_9 = \frac{\Delta t}{\Delta t_{R_9}} \left(\rho(c^4 + 27u_x^2 u_y^2) - 9u_x^2 a_6 - 9u_y^2 a_4 + 36u_x u_y a_3 \right) + 18u_x a_7 + 18u_y a_5 - 9a_8 - 0.5\Delta m_4 (4 + 3(u_x^2 + u_y^2)) + 4.5\Delta m_5 (u_x^2 + u_y^2) - 36\Delta m_6 u_x u_y - 6(\Delta m_7 u_y + \Delta m_8 u_x), \quad (10)$$

where

$$\begin{aligned} a_1 &= \langle c_x^2 + c_y^2 | f \rangle, & a_2 &= \langle c_x^2 - c_y^2 | f \rangle, \\ a_3 &= \langle c_x c_y | f \rangle, & a_4 &= \langle c_x^2 | f \rangle, \\ a_5 &= \langle c_x^2 c_y | f \rangle, & a_6 &= \langle c_y^2 | f \rangle, \\ a_7 &= \langle c_x c_y^2 | f \rangle, & a_8 &= \langle c_x^2 c_y^2 | f \rangle. \end{aligned}$$

Eqs. (2) to (10) were constructed using central moments of the finite distributions and of a continuous equilibrium distribution. The latter were obtained by integrating central moments of the Maxwell-Boltzmann distribution in the microscopic velocity space. The Maxwell-Boltzmann distribution can also be expanded around the macroscopic velocity space origin, and, after applying a Gauss-Hermite quadrature [20], yield the following finite equilibrium distribution equations:

$$f_{eq,i} = \rho w_i \left(1 + \frac{\vec{u} \cdot \vec{c}_i}{c_s^2} + \frac{1}{2} \frac{(\vec{u} \cdot \vec{c}_i)^2}{c_s^4} - \frac{1}{2} \frac{\vec{u} \cdot \vec{u}}{c_s^2} \right). \quad (11)$$

The weights, w_i , and the lattice discrete velocities, \vec{c}_i , are given in Eqs. (12) and (13).

$$w_i = \begin{cases} 4/9, & \text{for } i=1 \\ 1/9, & \text{for } i=2,\dots,5 \\ 1/36, & \text{for } i=6,\dots,9 \end{cases} \quad (12)$$

$$\vec{c}_i = (c_x, c_y)_i = c \begin{cases} (0,0), & \text{for } i=1 \\ \left(\cos\left((i-2)\frac{\pi}{2}\right), \sin\left((i-2)\frac{\pi}{2}\right) \right), & \text{for } i=2,\dots,5 \\ \left(\cos\left(\frac{\pi}{4} + (i-6)\frac{\pi}{2}\right), \sin\left(\frac{\pi}{4} + (i-6)\frac{\pi}{2}\right) \right), & \text{for } i=6,\dots,9 \end{cases} \quad (13)$$

The matrix \mathbf{M} in Eq. (1) is used to perform the transformation from distribution space to moment space [21],

$$|m\rangle = \mathbf{M} |f\rangle = \begin{bmatrix} \langle 1 | \\ \langle c_x | \\ \langle c_y | \\ 3\langle c_x^2 + c_y^2 | - 4\langle 1 | \\ \langle c_x^2 - c_y^2 | \\ \langle c_x c_y | \\ -3\langle c_x^2 c_y | + 2\langle c_y | \\ -3\langle c_x c_y^2 | + 2\langle c_x | \\ 9\langle c_x^2 c_y^2 | - 6\langle c_x^2 + c_y^2 | + 4\langle 1 | \end{bmatrix} |f\rangle. \quad (14)$$

The constant c is the lattice speed, $c=\Delta x/\Delta t$, and is related to the lattice speed of sound through

$$c_s^2 = \frac{c^2}{3}, \quad (15)$$

which serves to determine the relaxation time Δt_R with the following relation involving the lattice kinematic viscosity:

$$\nu = c_s^2 \Delta t \left(\frac{\Delta t_R}{\Delta t} - \frac{1}{2} \right). \quad (16)$$

The time step Δt is set equal to Δx such that c is equal to 1. In Eqs. (6) and (7), the relaxation times Δt_{R_5} and Δt_{R_6} are set as in Eq. (16). For the other moment variation equations, Geier [18] suggests that the cascaded model is typically stable with $\Delta t / \Delta t_{R_{4,7,8,9}}$ all set to unity. Premnath and Banerjee [21] indicate that it is also possible to choose to over-relax these moment variations in order to control numerical dissipation without losing on computational stability. We followed the second approach, however we did not perform a sensitivity analysis of the effect of varying these relaxation times on our simulation results. We simply chose to set $\Delta t / \Delta t_{R_4} = \Delta t / \Delta t_{R_9} = 1.4$ and

$\Delta t / \Delta t_{R_7} = \Delta t / \Delta t_{R_8} = 1.2$ following the work of Mezrhab et al. [22], even though they used these coefficients with a multiple-relaxation collision operator.

Distributions are propagated to the neighbor nodes through the usual LBM streaming step:

$$f_i(x + c_{x,i}\Delta t, y + c_{y,i}\Delta t, t + \Delta t) = f_i^*(x, y, t), \quad (17)$$

and the macroscopic properties correspond to the zeroth and first-order moments of the distributions:

$$\rho = \langle 1 | f \rangle, \quad \rho u_x = \langle c_x | f \rangle, \quad \rho u_y = \langle c_y | f \rangle. \quad (18)$$

As a final remark on the chosen LBM model, we must mention that there exist other collision operators developed for improving the computational stability over the conventional BGK model. For instance, Brownlee et al. [23] proposed the use of entropy limiters, and Latt and Chopard [24] introduced the regularized model. However, the cascaded operator seemed appropriate for our application since it has also been utilized for aerodynamic analyses by Brionnaud et al. [14].

3. Multi-Domain Algorithm

One advantage of the LBM is the ease with which boundary conditions are implemented over curved surfaces while the flow is solved over a regular lattice, or grid. However, as with any CFD method, solution accuracy increases with grid refinement, and the computational cost of using a fine regular grid over the entire domain is prohibitive. Many grid refinement techniques have been proposed for the LBM, starting with the work of Fillippova and Hänel [25], who introduced a method for transferring particle distribution information between superposed grids of different resolutions, based on an adjustment of non-equilibrium distributions. This adjustment depends on the relaxation parameters of each grid level, which are unique and set to ensure that the viscosity and Reynolds number are equal for all levels. In recent work by Lagrava et al. [26], a method of the same type is used, and distribution filtering is introduced in the information transfer from fine to coarse grids. They obtained accurate solutions for a low Reynolds cylinder flow and a dipole wall collision benchmark case. This is the method we are using here. We also follow the advice of Lagrava et al. pertaining to turbulent flows, which is that the domain overlap region be larger than one coarse grid spacing. This is illustrated in Fig. 1.

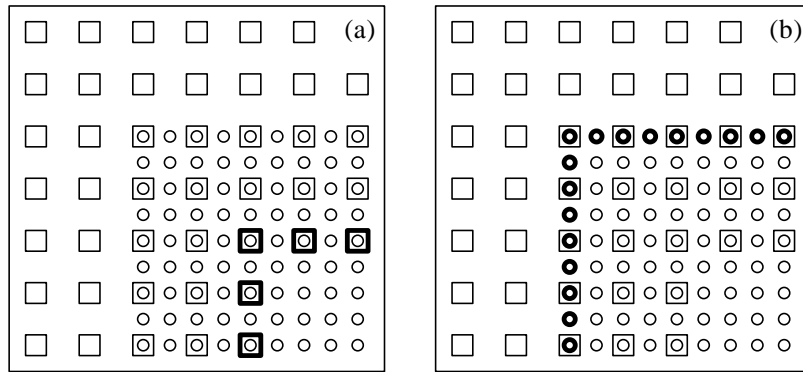


Fig. 1. Example of the overlap area between domains of different grid spacings. The thick symbols indicate the lattice sites onto which distribution information is transferred from the adjacent domain: a) fine to coarse, and b) coarse to fine.

In this method, all boundary distributions are replaced, not only those that are unknown after streaming. The general implementation of this method is detailed in the algorithm below.

- Step 0 – Initialize ρ , u_x , and u_y on all domains, and use their values to initialize the equilibrium distributions. Set all distributions to their equilibrium values.
- Step 1 – Spatially interpolate and record coarse distributions at the beginning of the coarse time step to be used in the coarse-to-fine temporal interpolation.
- Step 2 – Perform the collision on the coarse domain.
- Step 3 – Perform the streaming on the coarse domain.
- Step 4 – Calculate ρ , u_x , and u_y on the coarse domain.
- Step 5 – Calculate the equilibrium distributions on the coarse domain.
- Step 6 – Spatially interpolate and record coarse distributions at the end of the coarse time step to be used in the coarse-to-fine temporal interpolation.
- Step 7 – Perform the collision on the fine domain.
- Step 8 – Perform the streaming on the fine domain.
- Step 9 – Calculate ρ , u_x , and u_y on the fine domain.
- Step 10 – Calculate the equilibrium distributions on the fine domain.
- Step 11 – Perform coarse-to-fine temporal distribution interpolation, as well as scaling.
- Step 12 – Update ρ , u_x , and u_y on the fine boundary lattice sites.
- Step 13 – Repeat Steps 7 to 12, then go to Step 14.
- Step 14 – Transfer the distributions from the fine domain to the coarse domain, using filtering and scaling.
- Step 15 – Update ρ , u_x , and u_y on the coarse boundary lattice sites.
- Step 16 – Go back to Step 1.

The filtering process in Step 14 is applied to both nominal and equilibrium distributions, and uses 9 neighboring fine lattice sites:

$$f_i^{filtered} = \frac{1}{4} f_i + \frac{1}{8} (f_i^E + f_i^N + f_i^W + f_i^S) + \frac{1}{16} (f_i^{NE} + f_i^{NW} + f_i^{SW} + f_i^{SE}). \quad (19)$$

The spatial interpolation process in Steps 1 and 6 is cubic, and uses a stencil of 4 coarse lattice sites along the boundary. The temporal interpolation in Step 11 is linear, and is executed after each collide-and-stream cycle of the fine domain. Since the coarse and fine domains have a different relaxation coefficient $\Delta t / \Delta t_R$, the interpolated and filtered distributions in Steps 11 and 14 must be scaled [26]:

$$f_i^{coarse} = \left(1 - \Delta t_{R_c} / \Delta t_{R_f}\right) f_{eq,i}^{fine,filtered} + \left(\Delta t_{R_c} / \Delta t_{R_f}\right) f_i^{fine,filtered}, \quad (20)$$

$$f_i^{fine} = \left(1 - \Delta t_{R_f} / \Delta t_{R_c}\right) f_{eq,i}^{coarse,interpolated} + \left(\Delta t_{R_f} / \Delta t_{R_c}\right) f_i^{coarse,interpolated}. \quad (21)$$

Since the equilibrium distributions depend on macroscopic properties, they do not require any scaling:

$$f_{eq,i}^{coarse} = f_{eq,i}^{fine,filtered}, \quad f_{eq,i}^{fine} = f_{eq,i}^{coarse,interpolated}. \quad (22)$$

Fig. 2 illustrates a typical multi-domain grid refinement over an airfoil of 1 m chord which is used in this work. The thickness of the domains is set to approximately 12 grid spacings.

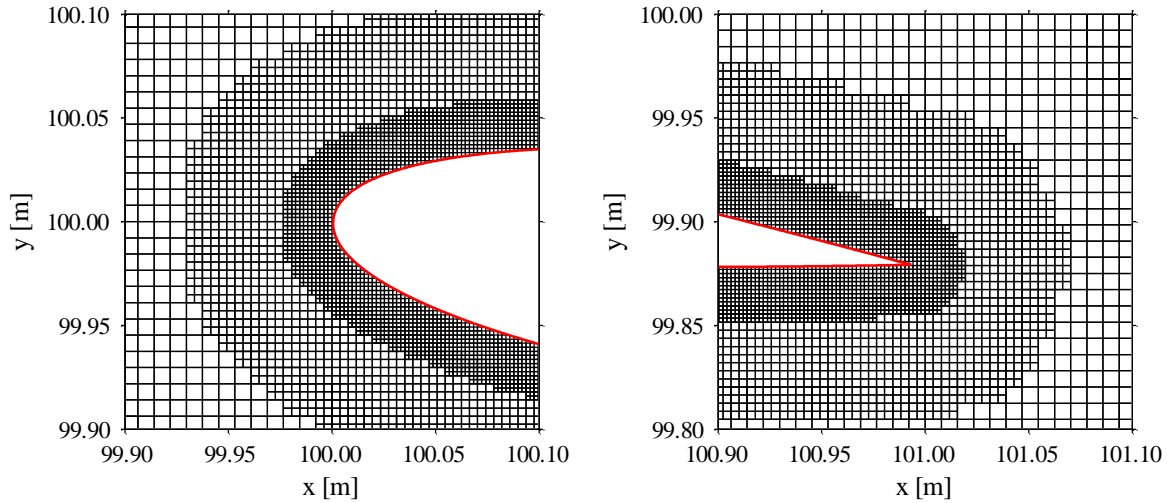


Fig. 2. Representation of a typical multi-domain geometry over the leading and trailing edges of a NACA 0012 airfoil of 1m chord at 7° angle of attack. This example uses ten domain levels, and the coarsest grid spacing is 512 times larger than the finest.

In our computer code, all the steps of the general algorithm are actually implemented through a recursive function that allow us to use any number of domains. This function is represented by the following pseudo-code.

```
function Main
    Nlevels=10; //number of domains
    for t=1 to Nt //loop over all coarse time steps
        Level=1; //current domain
        MultiDomain(Nlevels,Level);
    end for
end function

function MultiDomain(Nlevels,Level)
    if(Level<Nlevels)
        RecordStartDistributions(Level);
        Collision(Level);
        Streaming(Level);
        PhysicalProperties(Level);
        EquilibriumDistributions(Level);
        RecordEndDistributions(Level);
        MultiDomain(Nlevels,Level+1);
        GridTransferCoarseToFine(Level);
        MultiDomain(Nlevels,Level+1);
        GridTransferCoarseToFine(Level);
        GridTransferFineToCoarse(Level+1);
        if(Level=1)
            DomainBoundaryConditions;
        end if
    else // (Level=Nlevels)
        Collision(Level);
        Streaming(Level);
        AirfoilBoundaryConditions;
```

```

PhysicalProperties(Level);
EquilibriumDistributions(Level);
end if
end function

```

4. Spalart-Allmaras Turbulence Model

The Spalart-Allmaras turbulence model [12] is a one-equation model that solves a transport equation for a viscosity quantity $\tilde{\nu}$, and uses this quantity to modulate a turbulent viscosity ν_t . In an LBM framework, the turbulent viscosity can then be used to modify the relaxation time in the collision operator. Here, we implement a modified version of the model, introduced by Edwards and Chandra [27], to improve the near wall numerical behavior in the laminar sublayer. The transport equation and the full set of parameters involved in the model are provided below:

$$\underbrace{\frac{\partial \tilde{\nu}}{\partial t} + \vec{u} \cdot (\vec{\nabla} \tilde{\nu})}_{\text{Advection}} = \underbrace{\frac{1}{\sigma} \left[\vec{\nabla} \cdot ((\nu + \tilde{\nu}) \vec{\nabla} \tilde{\nu}) + c_{b2} (\vec{\nabla} \tilde{\nu})^2 \right]}_{\text{Diffusion}} + \underbrace{c_{b1} \tilde{S} \tilde{\nu}}_{\text{Production}} - \underbrace{c_{w1} f_w \left(\frac{\tilde{\nu}}{d} \right)^2}_{\text{Destruction}}, \quad (23)$$

$$\nu_t = \tilde{\nu} f_{v1}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu}, \quad (24)$$

$$\tilde{S} = S^{1/2} \left[\frac{1}{\chi} + f_{v1} \right], \quad (25)$$

$$f_w = g \left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right]^{1/6}, \quad g = r + c_{w2} (r^6 - 6), \quad r = \frac{\tanh \left[\tilde{\nu} / (\tilde{S} \kappa^2 d^2) \right]}{\tanh(1.0)}, \quad (26)$$

$$c_{b1} = 0.1355, \quad \sigma = 2/3, \quad c_{b2} = 0.622, \quad \kappa = 0.41, \quad (27)$$

$$c_{w1} = c_{b1} / \kappa^2 + (1 + c_{b2}) / \sigma, \quad c_{w2} = 0.3, \quad c_{w3} = 2, \quad c_{v1} = 7.1. \quad (28)$$

In the original model, S represents the vorticity magnitude, whereas in the SA-Edwards model it is defined as

$$S = \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \frac{\partial u_i}{\partial x_j} - \frac{2}{3} \left(\frac{\partial u_k}{\partial x_k} \right)^2, \quad (29)$$

where the Einstein summation rule applies. The constant d is the distance to the closest wall. On the walls, the imposed boundary condition is $\tilde{\nu} = 0$.

Numerical implementation

The above set of equations is implemented in our code through a finite-difference scheme in both space and time, and, since the nominal kinematic viscosity appearing in the diffusion term is constant, Eq. (23) is modified to

$$\tilde{v}^{t+\Delta t} = \tilde{v}^t + \Delta t \left[-\vec{u}^t \cdot \vec{\nabla} \tilde{v}^t + \left(\frac{\nu + \tilde{v}^t}{\sigma} \right) \nabla^2 \tilde{v}^t + \left(\frac{c_{b2} + 1}{\sigma} \right) (\vec{\nabla} \tilde{v}^t)^2 + c_{b1} \tilde{S} \tilde{v}^t - c_{w1} f_w \left(\frac{\tilde{v}^t}{d} \right)^2 \right]. \quad (30)$$

The spatial derivatives of \tilde{v}^t , u_x^t , and u_y^t used in Eqs. (29) and (30) are provided in Eqs. (31) and (32), where s represents either of these three scalars.

$$\frac{\partial s}{\partial x} = \begin{cases} \frac{s_x - s_{x-\Delta x}}{\Delta x}, u_x^t \geq 0 \\ \frac{s_{x+\Delta x} - s_x}{\Delta x}, u_x^t < 0 \end{cases}, \quad \frac{\partial s}{\partial y} = \begin{cases} \frac{s_y - s_{y-\Delta y}}{\Delta y}, u_y^t \geq 0 \\ \frac{s_{y+\Delta y} - s_y}{\Delta y}, u_y^t < 0 \end{cases}, \quad (31)$$

$$\frac{\partial^2 s}{\partial x^2} = \frac{s_{x+\Delta x} - 2s_x + s_{x-\Delta x}}{\Delta x^2}, \quad \frac{\partial^2 s}{\partial y^2} = \frac{s_{y+\Delta y} - 2s_y + s_{y-\Delta y}}{\Delta y^2}. \quad (32)$$

The upwinding scheme of Eq. (31) is introduced in the first derivatives to reduce the spurious oscillation phenomenon associated with central differencing. The $\tilde{v} = 0$ boundary condition is imposed through the gradient calculations, for which the expressions of Eqs. (31) and (32) are modified near the walls to account for the varying distances from the nearest lattice sites.

Spalart and Allmaras [12] indicate that it is best to impose $\tilde{v} = 0$ in the free stream, and that $\tilde{v} < \nu / 10$ is also acceptable. Imposing a non null free stream value is actually necessary to seed viscosity into the system, otherwise \tilde{v} would remain 0 everywhere. In our code, this is done with the following modification to the production term:

$$c_{b1} \tilde{S} \max(\tilde{v}^t, \nu / 100). \quad (33)$$

In order to avoid the production of negative turbulent viscosity, which would yield values of $\Delta t / \Delta t_R$ larger than 2, and therefore numerical instability in the flow solution, \tilde{v} is clipped to the positive range after solving Eq. (30).

The SA-Edwards model is implemented in our multi-domain algorithm in the following manner:

- ν_t , $\frac{\partial \tilde{v}}{\partial x}$, $\frac{\partial \tilde{v}}{\partial y}$, $\nabla^2 \tilde{v}$, and S are calculated at the same time as the other physical properties, in Steps 4 and 9.
- Eq. (30) is solved after the equilibrium distribution calculations, and Steps 5.1 and 10.1 are introduced for that purpose.
- Steps 1 and 6, which record distributions for the coarse-to-fine temporal and spatial interpolations, are updated to record values of ν_t , $\frac{\partial \tilde{v}}{\partial x}$, $\frac{\partial \tilde{v}}{\partial y}$, $\nabla^2 \tilde{v}$, and S as well. Step 11 is also updated to perform the interpolation of these quantities.
- Step 14 is updated to perform the fine-to-coarse filtering of these quantities.

After Eq. (30) has been solved and when the turbulent viscosity, ν_t , has been transferred from one domain to another, the relaxation coefficient $\Delta t / \Delta t_R$ is updated with

$$\nu + \nu_t = c_s^2 \Delta t \left(\frac{\Delta t_R}{\Delta t} - \frac{1}{2} \right). \quad (34)$$

5. Boundary Condition on the Airfoil

A modified version of the velocity boundary condition introduced by Pellerin et al. [28] is used at the airfoil surface. This method is based on the streaming of equilibrium distributions, emanating from ghost nodes, the location of which depends on the relaxation coefficient used in the collision operator, $\Delta t/\Delta t_R$. After the streaming step, the fluid nodes located near a solid boundary are missing one or more distributions. These missing distributions are streamed from the ghost nodes located at a distance $q = \Delta x \Delta t_R / \Delta t$ from the fluid nodes, and are calculated with

$$f_i^{eq} = \rho_a w_i \left(1 + \frac{\vec{c}_i \cdot \vec{u}_g}{c_s^2} + \frac{1}{2} \frac{(\vec{c}_i \cdot \vec{u}_g)^2}{c_s^4} - \frac{1}{2} \frac{(\vec{u}_g \cdot \vec{u}_g)}{c_s^2} \right), \quad (35)$$

where \vec{u}_g is the pre-streaming velocity at the ghost node, and ρ_a is the pre-streaming density linearly interpolated at point a , as shown in Fig. 3. In the original version of the method, the ghost node velocity was established with an interpolation or extrapolation of the velocities linearly interpolated at two points in the flow field and the boundary velocity. In the method presented here, only one flow field interpolation point is used. This change was introduced to simplify the code implementation of the method. Eqs. (36) and (37) provide the set of operations for obtaining \vec{u}_g .

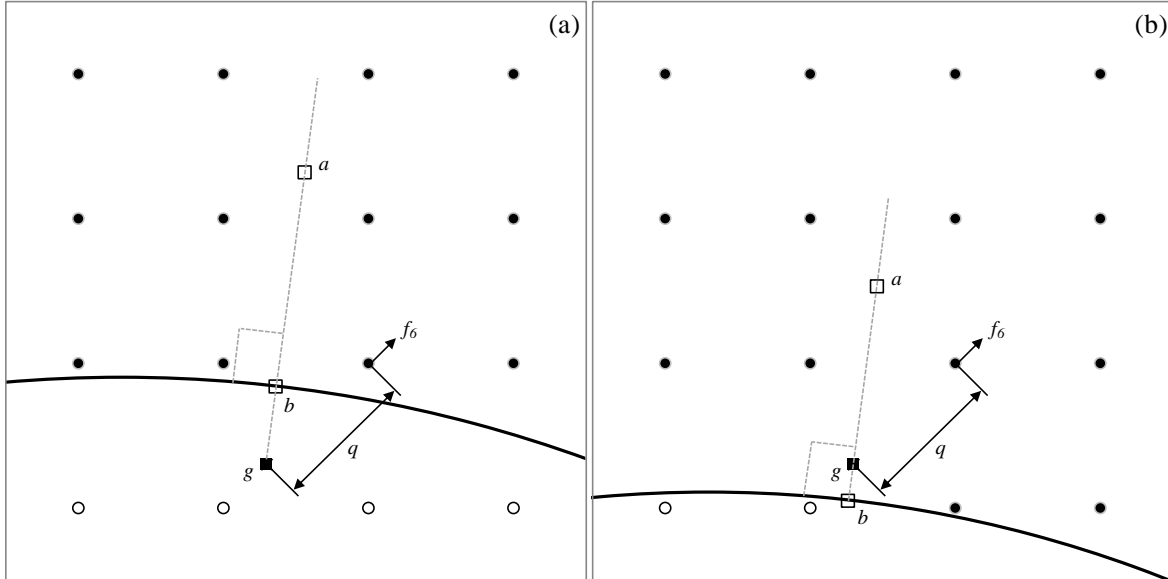


Fig. 3. Schematic of equilibrium distribution-based velocity boundary condition. Pane a) shows an extrapolation case, and pane b) shows an interpolation case.

$$\vec{u}_g = \vec{u}_a + k(\vec{u}_b - \vec{u}_a) \quad (36)$$

$$k = \frac{|\vec{x}_g - \vec{x}_a|}{|\vec{x}_b - \vec{x}_a|} \quad (37)$$

The distance between points a and b has been set to the same value for all ghost nodes, and chosen such that no coefficient k would be larger than 1.5 anywhere around the airfoil. This prevents the

introduction of flow solution instabilities caused by far distance extrapolations. For the flows studied, a distance of $1.5\Delta x$ was large enough to meet the chosen k limit.

Also, since we are simulating turbulent wall bounded flows, and because the process for calculating \vec{u}_g involves a linear operation, it is important that Δx be small enough for point a to be located close to the viscous sublayer of the turbulent boundary layer.

In this work, the curved boundary condition is applied over a sharp trailing edge NACA0012 airfoil [29], which is defined by

$$y/l = \pm 0.594689181 \left[\begin{aligned} &0.298222773 \text{sqrt}(x/l) - 0.127125232(x/l) \\ &- 0.357907906(x/l)^2 + 0.291984971(x/l)^3 \\ &- 0.105174606(x/l)^4 \end{aligned} \right], \quad (38)$$

and where l is the chord length. The letter l was chosen over the letter c to avoid any confusion with the lattice speed. The curved boundary condition is also applied over two non-symmetrical airfoils: the high lift profile, S1223, depicted in Fig. 4, and the E387 profile, depicted in Fig. 5.

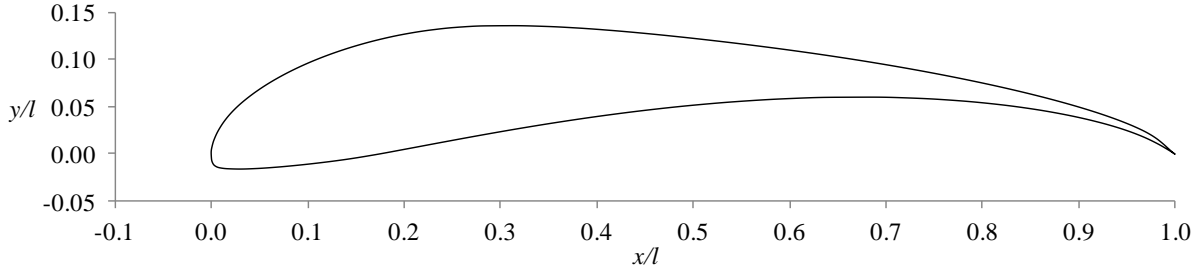


Fig. 4. S1223 profile [30].

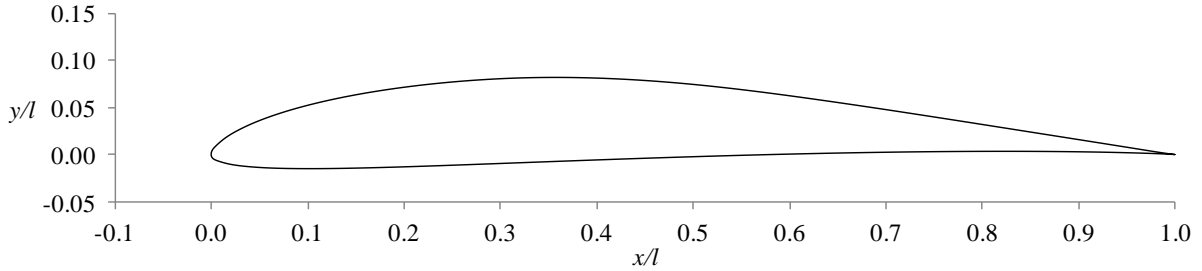


Fig. 5. E387 profile [30].

Even though we selected the curved boundary condition method of Pellerin et al. [28] for the present work, it is important to point out to the interested reader that there exist alternative choices, including the commonly used interpolated bounce-back scheme of Bouzidi et al. [31]. This method differs from the one used herein inasmuch as the post-streaming missing distributions are replaced by opposite direction distributions interpolated along neighboring lattice sites, whereas the current method interpolates macroscopic properties to reconstruct the missing distributions. While both methods are second-order accurate in space [28], the main advantage of the Bouzidi et al. method lies in its simpler numerical implementation. Conversely, we did not select it because it was reported not to conserve mass [32,33], and numerical results for the lid-driven cavity flow benchmark case showed that its produced velocity field is less accurate than for the method of Pellerin et al. when a boundary velocity is imposed. While only no-slip airfoil boundaries are used

in this work, our choice was also motivated by the future intention to use slip boundaries for the implementation of wall functions. That being said, given that the near boundary grid we use is very fine, we anticipate that had we used the Bouzidi et al. method for the current work, the resulting velocity field would have been similar.

6. Domain Boundary Conditions

The computational domain is square, with sides 200 airfoil chords long. The airfoil's leading edge is positioned in the center of the domain, as shown in Fig. 2. The domain boundary conditions are imposed in a two-step process. The first step consists of approximating the missing post-streaming distributions for all the lattice sites located on the perimeter of the domain. Since some of their streaming source lattice sites are located outside the domain, and therefore are non-existent, the corresponding missing post-streaming distributions are obtained by moving the source lattice sites to the closest sites in the domain. This approximation was taken from Pellerin et al. [28]. In the second step, the approximated distributions are modified to impose $u_y=0$ on the top and bottom boundaries, and $u_x=U_0$ and $\rho=\rho_0$ on the inlet. No further modifications are applied to the exit boundary.

Eqs. (39) to (44) provide the details for the second step. They use the following notations; $f_A = f_2 + f_6 + f_9$, $f_B = f_1 + f_3 + f_5$, $f_C = f_4 + f_7 + f_8$, $f_D = f_3 + f_6 + f_7$, $f_E = f_1 + f_2 + f_4$, and $f_F = f_5 + f_8 + f_9$. For the top boundary, $u_y=0$ is imposed through

$$\begin{bmatrix} \Delta f_5 \\ \Delta f_8 \\ \Delta f_9 \end{bmatrix} = (f'_F - f_F) \begin{bmatrix} 2/3 \\ 1/6 \\ 1/6 \end{bmatrix}, \quad (39)$$

where

$$f'_F = \frac{c f_D - u_y (f_D + f_E)}{u_y + c}. \quad (40)$$

Similarly, on the bottom boundary $u_y=0$ is imposed through

$$\begin{bmatrix} \Delta f_3 \\ \Delta f_6 \\ \Delta f_7 \end{bmatrix} = (f'_D - f_D) \begin{bmatrix} 2/3 \\ 1/6 \\ 1/6 \end{bmatrix}, \quad (41)$$

where

$$f'_D = \frac{c f_F + u_y (f_E + f_F)}{c - u_y}. \quad (42)$$

At the inlet, $u_x=U_0$ and $\rho=\rho_0$ cannot be imposed exactly and simultaneously, therefore an average correction is used:

$$\begin{bmatrix} \Delta f_2 \\ \Delta f_6 \\ \Delta f_9 \end{bmatrix} = \left(0.5 (f'_{A,u_x} + f'_{A,\rho}) - f_A \right) \begin{bmatrix} 2/3 \\ 1/6 \\ 1/6 \end{bmatrix}, \quad (43)$$

where

$$f'_{A,u_x} = \frac{c f_C + u_x (f_B + f_C)}{c - u_x}, \text{ and } f'_{A,\rho} = \rho - f_B - f_C. \quad (44)$$

Although u_x and ρ are not imposed exactly to the desired values, for a steady flow they will converge towards these values. The objective of imposing the density at the inlet is to reduce pressure wave reflections occurring when imposing density, or pressure, at the exit. The top and bottom domain corners at the inlet require a special treatment to impose u_y , u_x , and ρ at the same time. The procedure is not detailed here for sake of simplicity but essentially relies on solving a set of equations that take into account the coupling effects of the distribution variations, e.g. on the top left corner we solve for and impose Δf_A and Δf_F .

7. Flow Over a NACA0012 Airfoil

The proposed airfoil flow solution methodology is validated at a Reynolds number of 5×10^5 , and we compare our results with those of Lockard et al. [8], obtained with the CFL3D code, and Li et al. [11], obtained with the LBM. CFL3D solves the Reynolds-averaged Navier-Stokes equations with a finite-volume formulation, and permits to use various turbulence models. Lockard et al. [8] selected the Spalart-Allmaras model and integrated it down to the wall, and so did not use a wall function. They used a C-mesh of 373×141 grid spacings, or 52593 cells, which yielded a normal grid spacing of 0.00012 of a chord near the middle chord. In their opinion, this level of mesh refinement is sufficient to resolve the boundary layer. Li et al. [11] also use the Spalart-Allmaras model, in conjunction with a non-uniform mesh with a finest dimension of 0.0002 of a chord, which allowed them to obtain pressure coefficients and velocity profiles that compare reasonably well with the data of Lockard et al. [8] for angles of attack of 0° and 7° .

We use a multi-domain configuration that provides a grid spacing of $1/8192$ of a chord near the airfoil, which is comparable to the mesh refinement used for the numerical experiments of Lockard et al. [8]. Our lattice includes a total of approximately 550 thousand sites, with small variations depending on the angle of attack of the airfoil. This is considerably less than the 4 million sites of Li et al.'s [11] non-uniform mesh, which provides a finest refinement of 0.0002 of a chord. We ran our simulations until convergence of the force coefficients to the fourth decimal place was attained, and that occurred only after numerous time steps. For instance, for the largest angle of attack tested, 1000 iterations on the coarsest domain were necessary, which represents more than 8 million iterations on the finest domain. This required 10 days of CPU run time at a rate of approximately 4 coarse iterations per hour.

Below, we compare our results in terms of aerodynamic force coefficients, pressure coefficient profiles, and velocity profiles at various locations along the chord.

7.1 Aerodynamic Force Coefficients

Simulations were conducted for angles of attack of 0° , 3° , 7° , and 12° . The aerodynamic forces were computed with the momentum exchange method [34], and are provided in Table 1. The lift coefficient is also depicted in

Fig. 6(a) and the drag coefficient in Fig. 6(b). The experimental data of Jacobs and Sherman [35], digitized from Shock et al. [36], are also plotted. Even though they were obtained at a Reynolds

number of 6.6×10^5 , they serve as a qualitative means of comparison. We have found no experimental data close to $Re=5 \times 10^5$ in the recent literature.

AOA [°]	C_L		C_D	
	CFL3D	Present	CFL3D	Present
0	0.0000	0.0000	0.0128	0.0100
3	0.3237	0.3257	0.0130	0.0104
7	0.7449	0.7457	0.0157	0.0128
12	1.1809	1.1623	0.0275	0.0263

Table 1. Lift and drag coefficients for a NACA0012 airfoil at $Re=5 \times 10^5$.

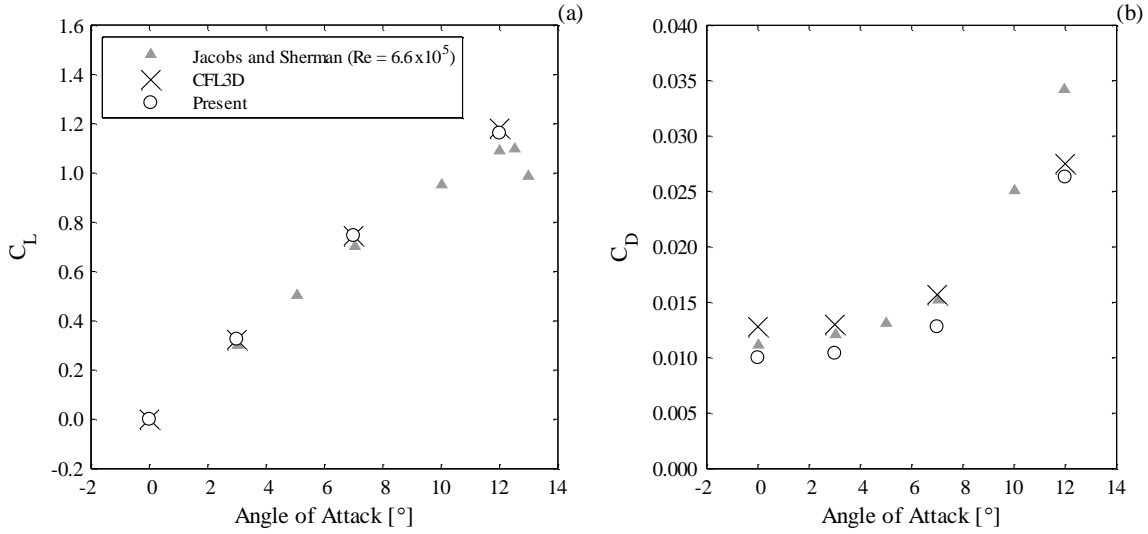


Fig. 6. Lift and drag coefficients for a NACA0012 airfoil at $Re=5 \times 10^5$.

In Fig. 7, we present the evolution of the force coefficients as a function of grid refinement, where N represents the number of grid spacings covering the airfoil chord. There is only one point for the 12° angle of attack at the chosen nominal grid refinement of $N=8192$, because coarser grid spacings produce detached flows.

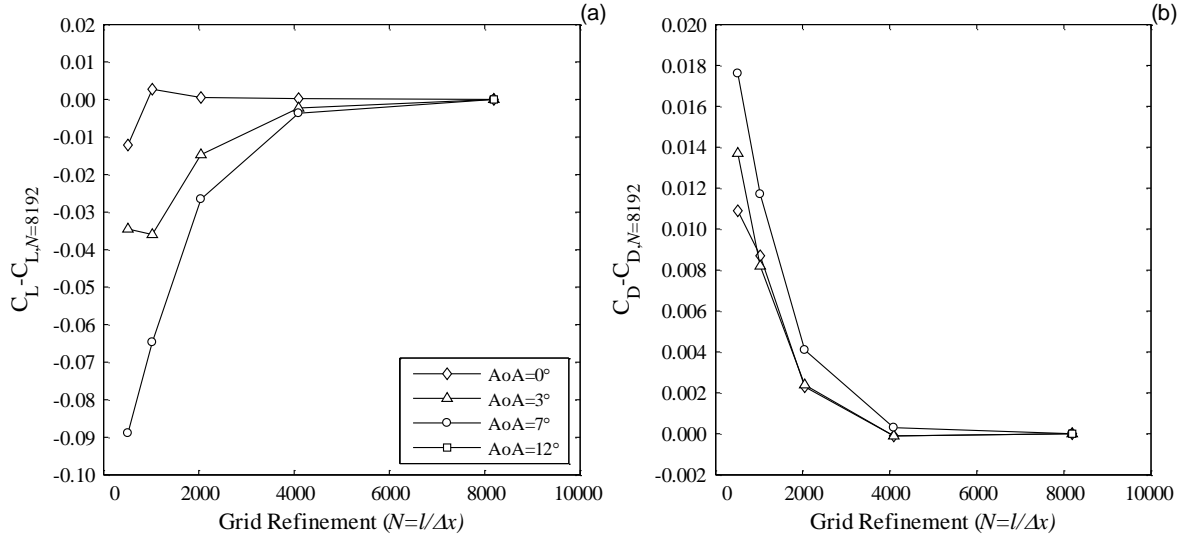


Fig. 7. Convergence of the lift and drag coefficients for a NACA0012 airfoil at $Re=5 \times 10^5$.

7.2 Pressure Coefficients

Figs. 8-10 display the pressure coefficients for angles of attack of 0° , 7° , and 12° . The x^*/l normalized coordinate is along the airfoil's centerline, with its origin at the leading edge of the profile.

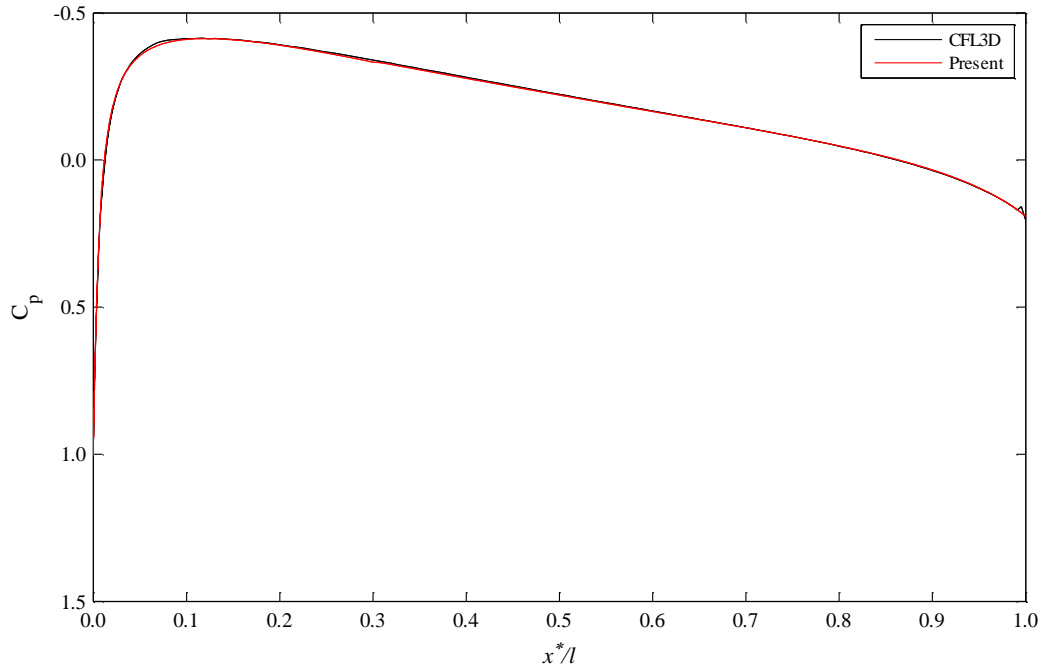


Fig. 8. Pressure coefficients for an angle of attack of 0° .

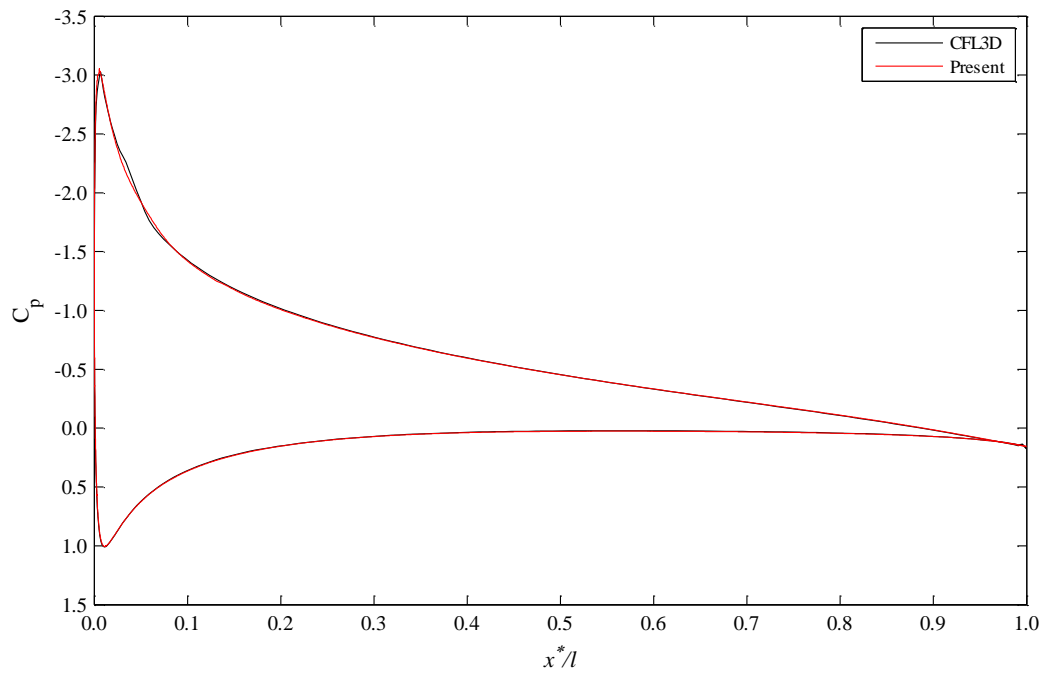


Fig. 9. Pressure coefficients for an angle of attack of 7° .

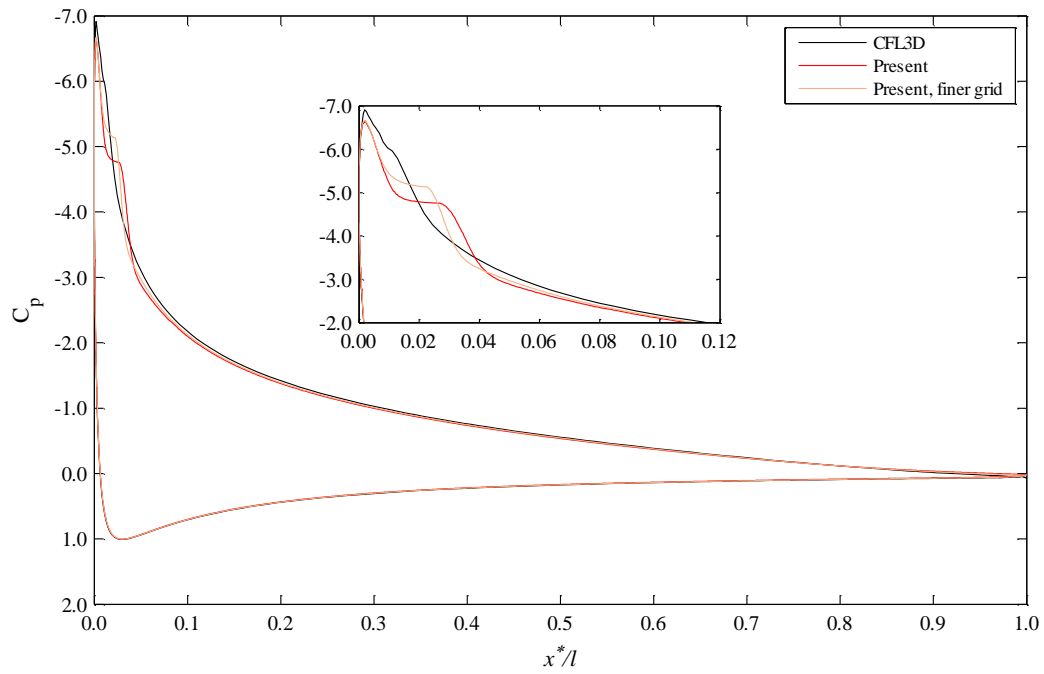


Fig. 10. Pressure coefficients for an angle of attack of 12° .

7.3 Velocity profiles

The normalized distance from the airfoil's centerline along its normal direction, y^*/l , is plotted against the normalized velocity component tangential to the airfoil centerline, u^*/U_∞ . The velocity profiles are provided for three different chord coordinates, x^*/l , for angles of attack of 0° and 7° in Fig. 11 and Fig. 12 respectively. Fig. 13 displays the velocity profile at $x^*/l=0$ for the 12° angle of attack.

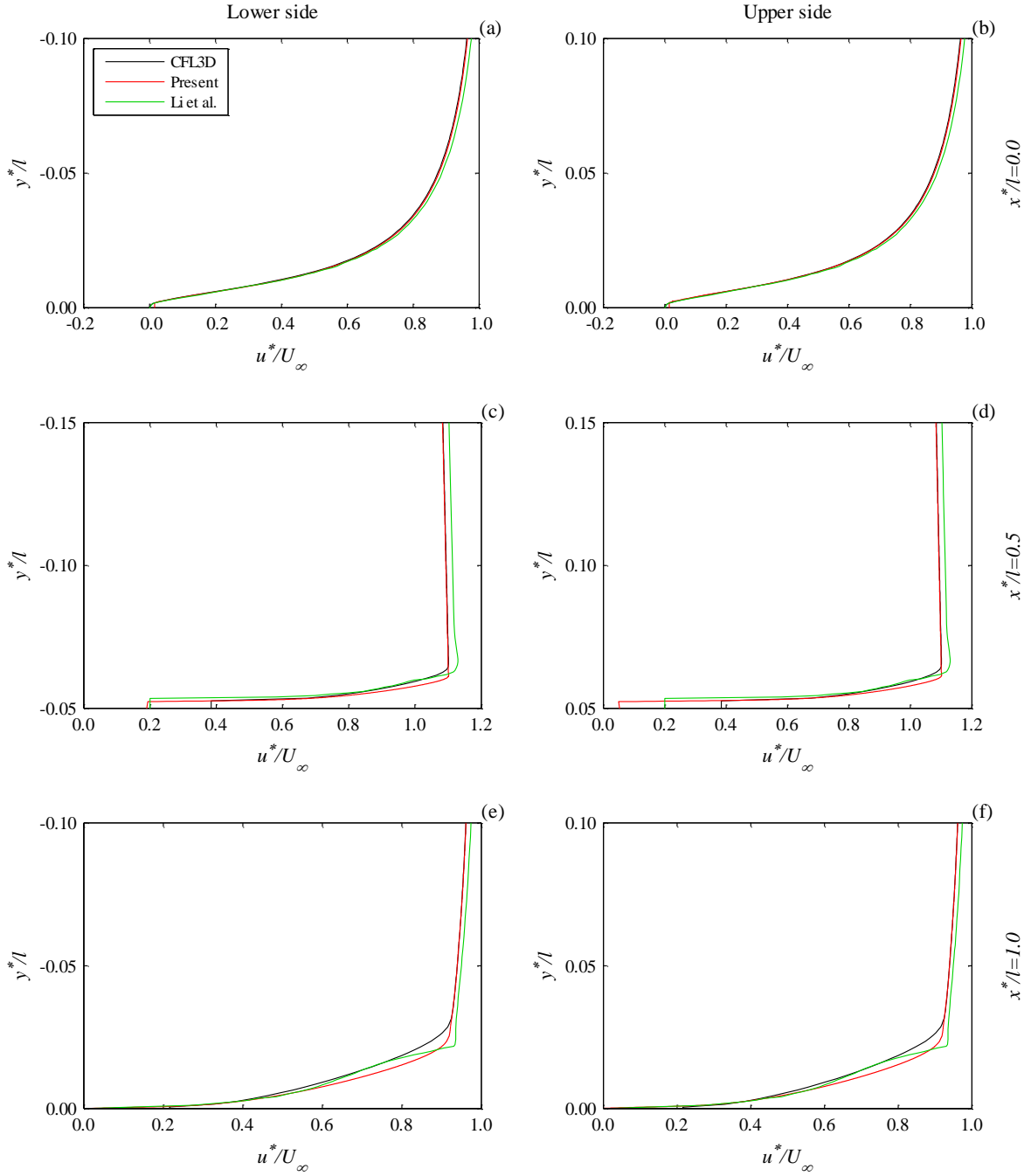


Fig. 11. Velocity profiles for a NACA0012 airfoil at 0° angle of attack at chord coordinates $x^*/l=0.0, 0.50$, and 1.0 .

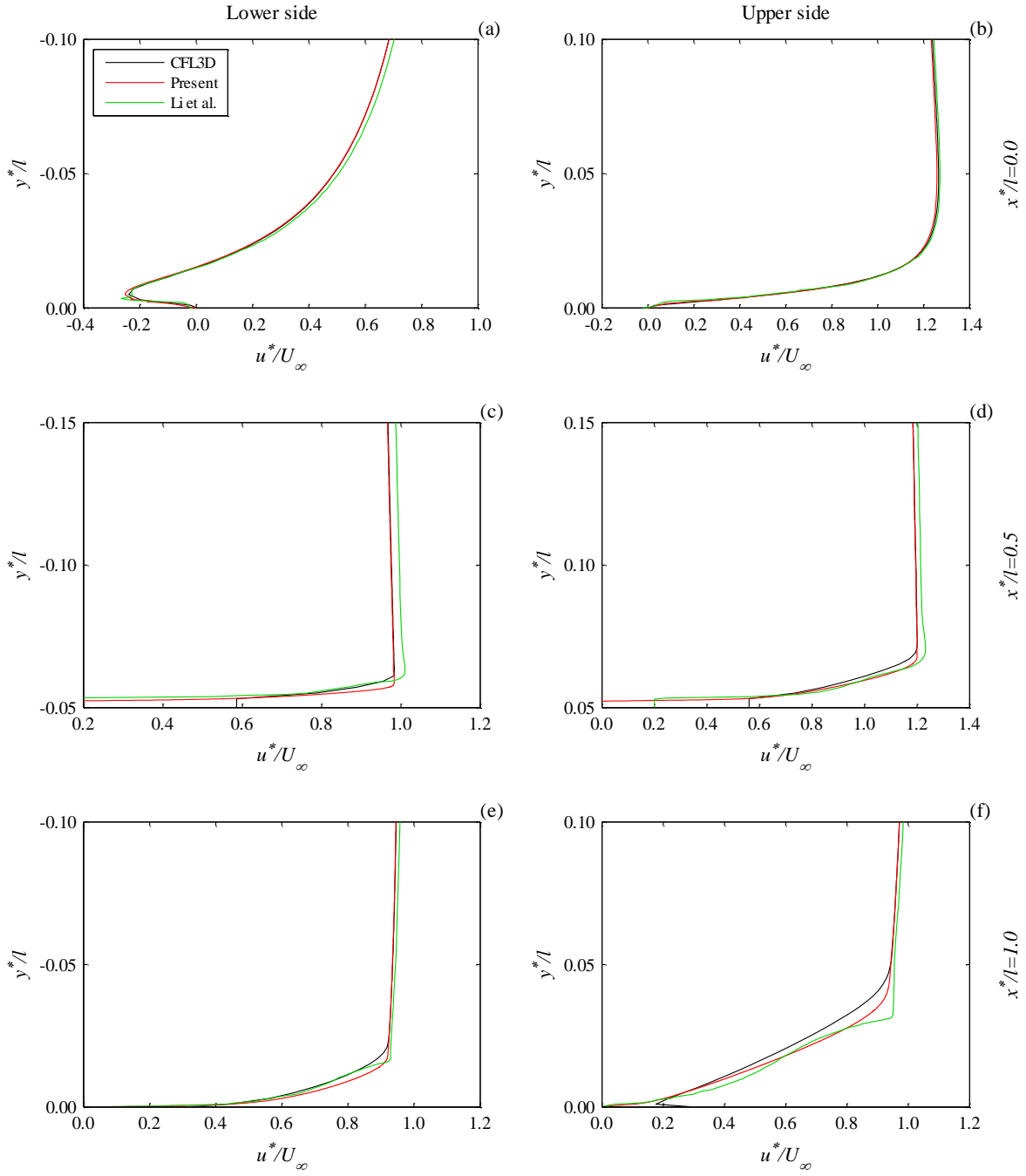


Fig. 12. Velocity profiles for a NACA0012 airfoil at 7° angle of attack at chord coordinates $x^*/l=0.0, 0.50$, and 1.0 .

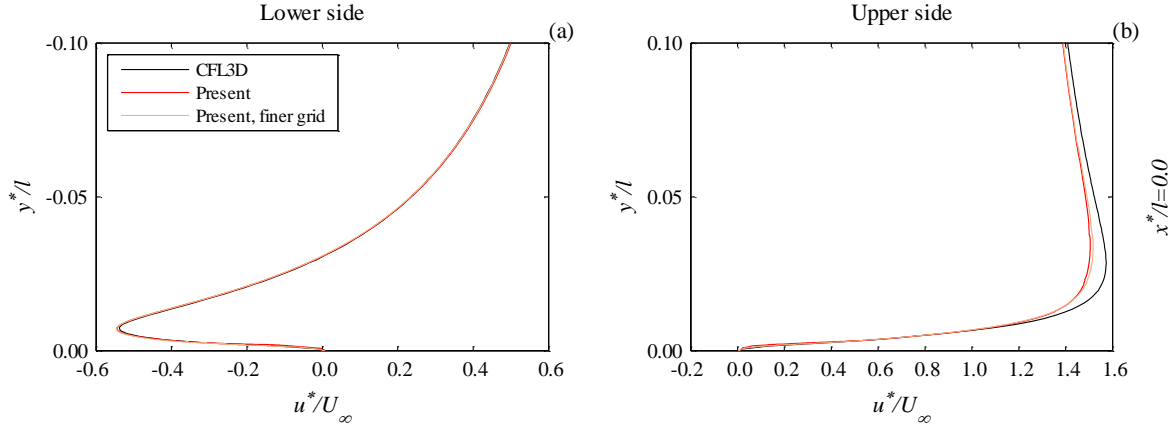


Fig. 13. Velocity profiles for a NACA0012 airfoil at 12° angle of attack at chord coordinate $x^*/l=0.0$.

8. Flow Over the S1223 and E387 Airfoils

In our second validation case, we solve the flow over two non-symmetrical airfoils, S1223 and E387, both at $Re=2 \times 10^5$. These airfoils were chosen for two reasons. First, they were designed for lower Reynolds numbers and therefore their aerodynamic characteristics are quite different from those of the NACA0012 airfoil, and second, experimental lift and drag coefficients are available in the literature for these airfoils [16,17]. Since the Re is lower than for the first validation case, we used a slightly coarser multi-domain grid, with a finest spacing of $\Delta x=l/4096$. This yielded a total of roughly 315 thousand sites for both airfoils for all the angles of attack tested. Force coefficient convergence for the near stall angles of attack was reached after 4 to 5 days of CPU run time at a rate of approximately 8 coarse iterations per hour. The resulting lift and drag coefficients for the S1223 and E387 airfoils are plotted in Fig. 14 and Fig. 15 respectively.

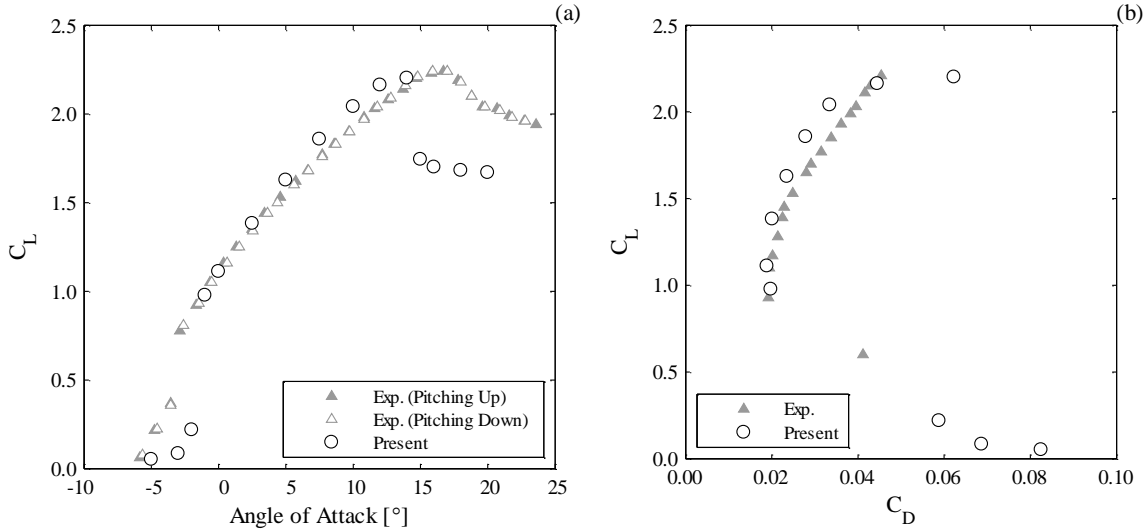


Fig. 14. Lift and drag coefficients for the S1223 airfoil at $Re=2 \times 10^5$.

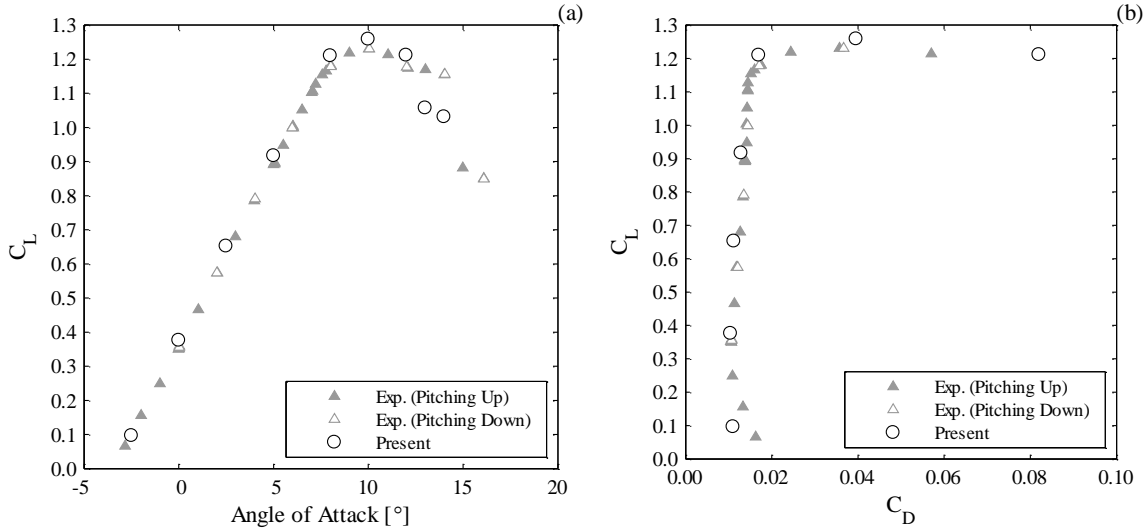


Fig. 15. Lift and drag coefficients for the E387 airfoil at $Re=2 \times 10^5$.

9. Discussion

For the first validation case, the lift coefficients produced with our method agree well with those obtained with CFL3D. There is a larger discrepancy at the 12° angle of attack, where our method predicts a coefficient that is approximately 0.02 smaller. The drag coefficient profile also compares well, although our method produces coefficients that are smaller by 24 drag counts on average. We cannot provide a formal explanation for this difference in drag, however some insight into the difference in lift at 12° can be found in the pressure coefficient profile depicted in Fig. 10. The dimple observed on the suction side of the airfoil near 0.02 chord reflects the laminar to turbulent boundary layer transition. In the CFL3D data, this dimple is also present, however it is closer to the leading edge and is much weaker. This indicates that the boundary layer is turbulent from earlier on. To verify whether our code could better capture this boundary layer transition location, and the associated effect on aerodynamic force coefficients, we ran a simulation with a finer grid, using a spacing of $1/16384$ of a chord near the airfoil. The simulation was run until convergence of the force coefficients to the third decimal place, and it produced $C_L=1.174$ and $C_D=0.025$. The associated velocity and pressure coefficient profiles are labelled *finer grid* in Fig. 13 and Fig. 10. The finer grid moved the boundary layer transition closer to the leading edge, and weakened its effect on the pressure coefficient profile. The lift coefficient is also in better agreement with the CFL3D result. The explanation for the difference in transition location presumably lies in the no-slip boundary condition. Near the leading edge, there is a large tangential laminar velocity gradient normal to the airfoil surface. In a cell-centered finite volume formulation, the flow velocity used for the no-slip boundary condition is located half a grid spacing from the boundary, whereas in our LBM boundary condition formulation this velocity is interpolated at one and a half grid spacings, therefore the former can better capture the velocity gradient near the wall when the grid spacings are comparable. In fact, our grid refinement verification moved the velocity interpolation points closer to the airfoil wall and the results were improved. This indicates that there is a need for further grid refinement in our methodology, or the introduction of a boundary slip velocity based on the assumed boundary layer velocity profile. This need is also supported by the grid convergence data presented in Fig. 7a, which show that, for the higher angles attack, lift coefficient convergence may not be fully reached, even at a finest grid spacing of $1/8192$ of a chord.

At angles of attack smaller than 12° , the pressure coefficients and velocity profiles compare very well with those obtained with CFL3D. In fact, the pressure coefficients are nearly indistinguishable. Our velocity profiles are slightly fuller than for CFL3D, however the profile shapes are similar. This observation is less applicable to the LBM results of Li et al. [11], particularly near the trailing edge, where their profile shapes differ from CFL3D. Our results appear to be in better agreement with the reference data, even after acknowledging that there is some uncertainty in the Li et al. [11] results, because we extracted them graphically from their article.

For the second validation case, our lift and drag coefficients for the E387 airfoil are in good agreement with the experimental data. The maximum lift is slightly over predicted, however the drag polar correctly captures the steep increase in drag near the maximum lift and stall onset. In our simulations, detached and unsteady flow occurred from 13° on, which indicates that stall is observed at a slightly lower angle of attack than in the experimental results. Nevertheless, there is good agreement over most of the attached flow portion of the lift curve. For the S1223 airfoil, the lift peak is predicted earlier than for the experimental results, and its value is slightly underpredicted. However, for the lower and positive angles of attack, our numerical results match fairly well with the experimental data, and that is also reflected in the drag polar. For the negative angles of attack, there is a noticeable difference in the lift curve, as the discontinuity is much steeper in our results than in the experimental results. At those angles, the flow starts to become detached over a section of the lower side of the airfoil, and the observed discrepancy is likely due to our model inaccurately capturing the length of this detached region.

As with the NACA0012 validation case, the lower lift obtained at higher angles of attack for the S1223 airfoil may be attributable to an insufficiently refined grid, particularly near the leading edge. This effect is more obvious for the S1223 airfoil than for the E387 airfoil, because the former is more cambered and induces a larger acceleration of the flow on the suction side.

Another interesting point of discussion concerns the grid refinement technique. According to Rohde et al. [37], grid refinement techniques based on a rescaling of the non-equilibrium distributions, such as the one we used, are restricted to the BGK collision operator. They also state that if a more sophisticated collision operator is applied, it would be preferable to use a grid refinement technique that is independent of the collision scheme. Obviously, we did not follow this guideline, but neither can we identify any adverse effect of not doing so on our solutions. In fact, the velocity profiles depicted in Figs. 11-13 span many domains and vary smoothly across the interfaces. Furthermore, Geller et al. [38] used the grid refinement technique of Yu et al. [39] and the multiple relaxation time collision operator to study laminar flows, and did not report any apparent difficulty associated with using this combination of methods.

It can be noticed that the coefficients used for rescaling the distributions in Eqs. (20) and (21) are dependent on viscosity, and that we do not modify them, even though the viscosity is modified by the Spalart-Allmaras turbulence model. A similar observation can be made for the airfoil boundary condition method, where the distance q depends on viscosity. In the latter case, we chose to use the nominal viscosity for a practical reason, which is that modifying q and the velocity interpolation points at every time step would be very costly computationally. However, a more rational justification can also be provided: near the boundary, the Spalart-Allmaras viscosity, $\tilde{\nu}$, is close to 0 by definition, and for high Reynolds number flows q tends towards $0.5\Delta x$ and will only

significantly differ from that value if $\tilde{\nu}$ is large. Therefore, the nominal viscosity approximation is valid. In contrast, for the distribution rescaling, $\tilde{\nu}$ can be large if a domain boundary is located in an area of high turbulence, and the rescaling coefficient can vary considerably. We conducted a few tests using variable rescaling coefficients, but did not observe significant changes in the aerodynamic results, and so we decided not to include this code modification for the simulation results presented here. This issue remains an open topic for more rigorous analysis.

Our final comment on the grid refinement concerns the choice of lattice Mach number and relaxation coefficient $\Delta t/\Delta t_R$ for each grid level. We set $\Delta t = \Delta x$ for each level, and the Mach number constant at $Ma = 0.1c_s$. This ensures that the ratio of time steps in physical units scales appropriately with the grid refinement ratio, i.e. a grid twice as fine has a physical time step that is half the size. With this approach, $\Delta t/\Delta t_R$ is different for all grid levels. According to Lätt [40], a better choice for simulating incompressible fluids would be to use a constant $\Delta t/\Delta t_R$ and a variable Ma . Progressively reducing Ma on finer grids would ensure that the LBM compressibility errors of $O(Ma^2)$ would reduce along the spatial discretization errors of $O(\Delta x^2)$. In our context, this method has one drawback, however, which is that a grid refined by a factor of 2 would have a physical time step that is smaller by a factor of 4, which ultimately makes the physical time step on the finest grid much smaller than for the constant Ma approach, assuming that the Ma is the same for both methods on the coarsest grid. For instance, with the grid configuration that we used for solving the NACA0012 flows, the finest time step would be 2^{13} times smaller. This implies that the simulation times would increase dramatically. The trade-off between lower compressibility errors and shorter run times is clearly in favor of the latter, especially since other factors such as boundary conditions and turbulence transport may have more impact on the results than the LBM compressibility errors.

10. Conclusion

We have proposed the use of the LBM with a multi-domain grid refinement, the cascaded collision operator, and the Spalart-Allmaras turbulence model to solve turbulent airfoil flows. We validated our methodology on a NACA0012 airfoil at a Reynolds number of 5×10^5 and four angles of attack for which the flow is known to remain attached. We compared our results in terms of force coefficients, pressure profiles, and velocity profiles to the numerical results of Lockard et al. [8] obtained with CFL3D. We also compared the velocity profiles for two angles of attack with the numerical results of Li et al. [11], produced with the LBM using a non-uniform mesh. Both these reference datasets were generated with the Spalart-Allmaras turbulence model. For low angles of attack, our results agreed well with the CFL3D data. For the highest angle of attack simulated, 12° , our lift coefficient was slightly smaller than that produced by CFL3D. We identified a lack of grid refinement as a contributing factor, and were able to improve our results by using a grid twice as fine near the airfoil wall. However, the refinement level that we used nominally was more adequate for the lower angles of attack. In fact, our velocity profiles compared better with CFL3D than those of Li et al. [11].

We also validated our methodology at $Re=2 \times 10^5$ on two other airfoils, the S1223 and the E387, both of which are non symmetrical. The lift and drag coefficients obtained agree well with experimental data, although we identified a potential need for further grid refinement for the S1223

airfoil at larger angles of attack, as was the case for the NACA0012 airfoil. Overall, the LBM method we proposed is shown to be adequate for solving turbulent airfoil flows. However, in order to limit the grid refinement required for solving the turbulent boundary layers, we simulated flows in the lower turbulent Reynolds range. Solving faster flows, in the order of 10^6 , would become impractical with our computational setup, which consisted of a Matlab code ran on a 3.6GHz Intel i7 CPU with 16GB of RAM. The total number of lattice sites would grow beyond a million, and simulation times would increase dramatically. In fact, one drawback we have identified with our method is the slow convergence and stabilization of the force coefficients, leading to long run times, in the order of days. One contributing factor to this slow convergence lies in the fact that our LBM implementation is an explicit time evolution of the modeled fluid dynamics, which is carried on until a steady-state solution is reached. This represents a disadvantage in comparison to finite-volume Navier-Stokes solvers such as CFL3D, because with those solvers the equations can be discretized and solved with implicit schemes. These schemes are generally more stable than their explicit counterparts and allow for the use of larger time step sizes, which reduces the number of iterations necessary for the solution to converge and leads to shorter run times. In fact, Lockard et al. [8] reported that the CFL3D run time for solving the flow over the NACA0012 airfoil at 12° was 6.3 hours. Even if a direct comparison is not appropriate since the codes were run on different machines, i.e. a PC vs an SGI workstation, our run times are significantly larger. Furthermore, Lockard et al. [8] indicate that the CFL3D convergence was accelerated with local time-stepping, grid sequencing, and multigriding techniques. Such features are not part of our code, which is at a research level state.

In light of the large run time drawback we have identified, future work will consist of including wall functions in the boundary condition scheme, which would make it possible to increase the Reynolds number while limiting the number of lattice sites. Finally, one advantage of the multi-domain LBM, which we mentioned in the introduction, is its ability to adapt to arbitrary shapes, including iced airfoils. Consequently, in future work we intend to test the proposed flow solving methodology over iced airfoils as well.

References

- [1] G. Eitel-Amor, M. Meinke, W. Schröder, A lattice-Boltzmann method with hierarchically refined meshes, *Comput. Fluids*. 75 (2013) 127–139.
- [2] K.N. Premnath, M.J. Pattison, S. Banerjee, Computation of transitional flow past a circular cylinder using multiblock lattice Boltzmann method with a dynamic subgrid scale model, *Fluid Dyn. Res.* 45 (2013) 55510.
- [3] J. Yang, E.S. Boek, A comparison study of multi-component Lattice Boltzmann models for flow in porous media applications, *Comput. Math. with Appl.* 65 (2013) 882–890.
- [4] S. Leclaire, N. Pellerin, M. Reggio, J.-Y. Trépanier, Enhanced equilibrium distribution functions for simulating immiscible multiphase flows with variable density ratios in a class of lattice Boltzmann models, *Int. J. Multiph. Flow*. 57 (2013) 159–168.

- [5] O. Filippova, S. Succi, F. Mazzocco, C. Arrighetti, G. Bella, D. Hänel, et al., Multiscale lattice Boltzmann schemes with turbulence modeling, *J. Comput. Phys.* 170 (2001) 812–829.
- [6] O. Filippova, D. Hänel, Boundary-fitting and local grid refinement for lattice-BGK models, *Int. J. Mod. Phys. C* 9 (1998) 1271–1279.
- [7] T. Imamura, K. Suzuki, T. Nakamura, M. Yoshida, Flow simulation around an airfoil by lattice Boltzmann method on generalized coordinates, *AIAA J.* 43 (2005) 1968–1973.
- [8] D.P. Lockard, L. Luo, S.D. Milder, B.A. Singer, Evaluation of PowerFLOW for aerodynamic applications, *J. Stat. Phys.* 107 (2002) 423–478.
- [9] Y. Li, R. Shock, R. Zhang, H. Chen, T.I.-P. Shih, Simulation of flow over an iced airfoil by using a lattice-Boltzmann method, in: 43rd AIAA Aerosp. Sci. Meet. Exhib., AIAA, Reno, NV, 2005: pp. 1–8.
- [10] H. Chen, C. Teixeira, K. Molvig, Realization of fluid boundary conditions via discrete Boltzmann dynamics, *Int. J. Mod. Phys. C* 09 (1998) 1281–1292.
- [11] K. Li, C. Zhong, C. Zhuo, J. Cao, Non-body-fitted Cartesian-mesh simulation of highly turbulent flows using multi-relaxation-time lattice Boltzmann method, *Comput. Math. with Appl.* 63 (2012) 1481–1496.
- [12] P.R. Spalart, S.R. Allmaras, A one-equation turbulence model for aerodynamic flows, in: 30th Aerosp. Sci. Meet. Exhib., AIAA, Reno, NV, 1992: pp. 1–22.
- [13] P. Spalart, Trends in turbulence treatments, in: Fluids 2000 Conf. Exhib., AIAA, Denver, CO, 2000: pp. 1–13.
- [14] R. Brionnaud, D.M. Holman, M. Chavez Modena, Aerodynamic analysis of the 2nd AIAA High Lift Prediction Workshop by a lattice-Boltzmann method solver, 31st AIAA Appl. Aerodyn. Conf. (2014).
- [15] D. Ricot, E. Foquet, H. Touil, E. Lévêque, H. Machrouki, F. Chevillotte, et al., Aeroacoustic computations with a new CFD solver based on the Lattice Boltzmann Method, in: Automot. Railr. Conf. Symp. - 7th Ed., SIA, Le Mans, France, 2012.
- [16] M.S. Selig, J.J. Guglielmo, High-lift low Reynolds number airfoil design, *J. Aircr.* 34 (1997) 72–79.
- [17] R.J. McGhee, B.S. Walker, B.F. Millard, Experimental results for the Eppler 387 airfoil at low Reynolds numbers in the Langley low-turbulence pressure tunnel, NASA, Technical Memorandum 4062, 1988.

- [18] M. Geier, A. Greiner, J.G. Korvink, Cascaded digital lattice Boltzmann automata for high Reynolds number flow., *Phys. Rev. E* 73 (2006) 066705.
- [19] S. Leclaire, N. Pellerin, M. Reggio, J.-Y. Trépanier, Multiphase flow modeling of spinodal decomposition based on the cascaded lattice Boltzmann method, *Phys. A Stat. Mech. Its Appl.* (2014).
- [20] X. He, L.-S. Luo, Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation, *Phys. Rev. E* 56 (1997) 6811–6817.
- [21] K. Premnath, S. Banerjee, Incorporating forcing terms in cascaded lattice Boltzmann approach by method of central moments, *Phys. Rev. E* 80 (2009) 036702.
- [22] A. Mezrhab, M. Amine, M. Jami, H. Naji, M. Bouzidi, Double MRT thermal lattice Boltzmann method for simulating convective flows, *Phys. Lett. A* 374 (2010) 3499–3507.
- [23] R.A. Brownlee, A.N. Gorban, J. Levesley, Nonequilibrium entropy limiters in lattice Boltzmann methods, *Phys. A Stat. Mech. Its Appl.* 387 (2008) 385–406.
- [24] J. Latt, B. Chopard, Lattice Boltzmann method with regularized non-equilibrium distribution functions, *Math. Comput. Simul.* 72 (2006) 165–168.
- [25] O. Filippova, D. Hänel, Grid refinement for lattice-BGK models, *J. Comput. Phys.* 147 (1998) 219–228.
- [26] D. Lagrava, O. Malaspinas, J. Latt, B. Chopard, Advances in multi-domain lattice Boltzmann grid refinement, *J. Comput. Phys.* 231 (2012) 4808–4822.
- [27] J.R. Edwards, S. Chandra, Comparison of eddy viscosity-transport turbulence models for three-dimensional, shock-separated flowfields, *AIAA J.* 34 (1996) 756–763.
- [28] N. Pellerin, S. Leclaire, M. Reggio, Equilibrium distributions for straight, curved, and immersed boundary conditions in the lattice Boltzmann method, *Comput. Fluids*. (2014) 126–135.
- [29] NASA, Turbulence modeling resource, 2D NACA 0012 airfoil validation case, http://turbmodels.larc.nasa.gov/naca0012_val.html, retrieved on Mar. 16th, 2014.
- [30] M.S. Selig, UIUC Airfoil Coordinates Database, http://m-selig.ae.illinois.edu/ads/coord_database.html#S, retrieved on Nov. 4th, 2014.
- [31] M. Bouzidi, M. Firdaouss, P. Lallemand, Momentum transfer of a Boltzmann-lattice fluid with boundaries, *Phys. Fluids* 13 (2012) 3452–3459.
- [32] P. Lallemand, L.-S. Luo, Lattice Boltzmann method for moving boundaries, *J. Comput. Phys.* 184 (2003) 406–421.

- [33] M. Rohde, D. Kandhai, J.J. Derksen, H.E.A. Van den Akker, Improved bounce-back methods for no-slip walls in lattice-Boltzmann schemes: Theory and simulations, *Phys. Rev. E.* 67 (2003) 066703.
- [34] R. Mei, D. Yu, W. Shyy, L.-S. Luo, Force evaluation in the lattice Boltzmann method involving curved geometry, *Phys. Rev. E.* 65 (2002) 041203.
- [35] E.N. Jacobs, A. Sherman, Airfoil section characteristics as affected by variations of the Reynolds number, Report No. 586, National Advisory Committee for Aeronautics, 1937.
- [36] R.A. Shock, S. Mallick, H. Chen, V. Yakhot, R. Zhang, Recent results on two-dimensional airfoils using a lattice Boltzmann-based algorithm, *J. Aircr.* 39 (2002) 434–439.
- [37] M. Rohde, D. Kandhai, J.J. Derksen, H.E.A. Van Den Akker, A generic, mass conservative local grid refinement technique for lattice-Boltzmann schemes, *Int. J. Numer. Methods Fluids.* (2006) 439–468.
- [38] S. Geller, M. Krafczyk, J. Tölke, S. Turek, J. Hron, Benchmark computations based on lattice-Boltzmann, finite element and finite volume methods for laminar flows, *Comput. Fluids.* 35 (2006) 888–897.
- [39] D. Yu, R. Mei, W. Shyy, A multi-block lattice Boltzmann method for viscous fluid flows, *Int. J. Numer. Methods Fluids.* 39 (2002) 99–120.
- [40] J. Lätt, Hydrodynamic limit of lattice Boltzmann equations, PhD Thesis, Université de Genève, 2007.

ANNEXE D – ARTICLE 3 : SOLVING INCOMPRESSIBLE FLUID FLOWS ON UNSTRUCTURED MESHES WITH THE LATTICE BOLTZMANN FLUX SOLVER

Cet article a été publié dans une revue scientifique avec comité de lecture. La version publiée diffère de la version soumise suite à des modifications effectuées par l'éditeur.

Pellerin N., Leclaire S., Reggio M. Solving incompressible fluid flows on unstructured meshes with the lattice Boltzmann flux solver. Eng Appl Comput Fluid Mech 2017;11:310–27.

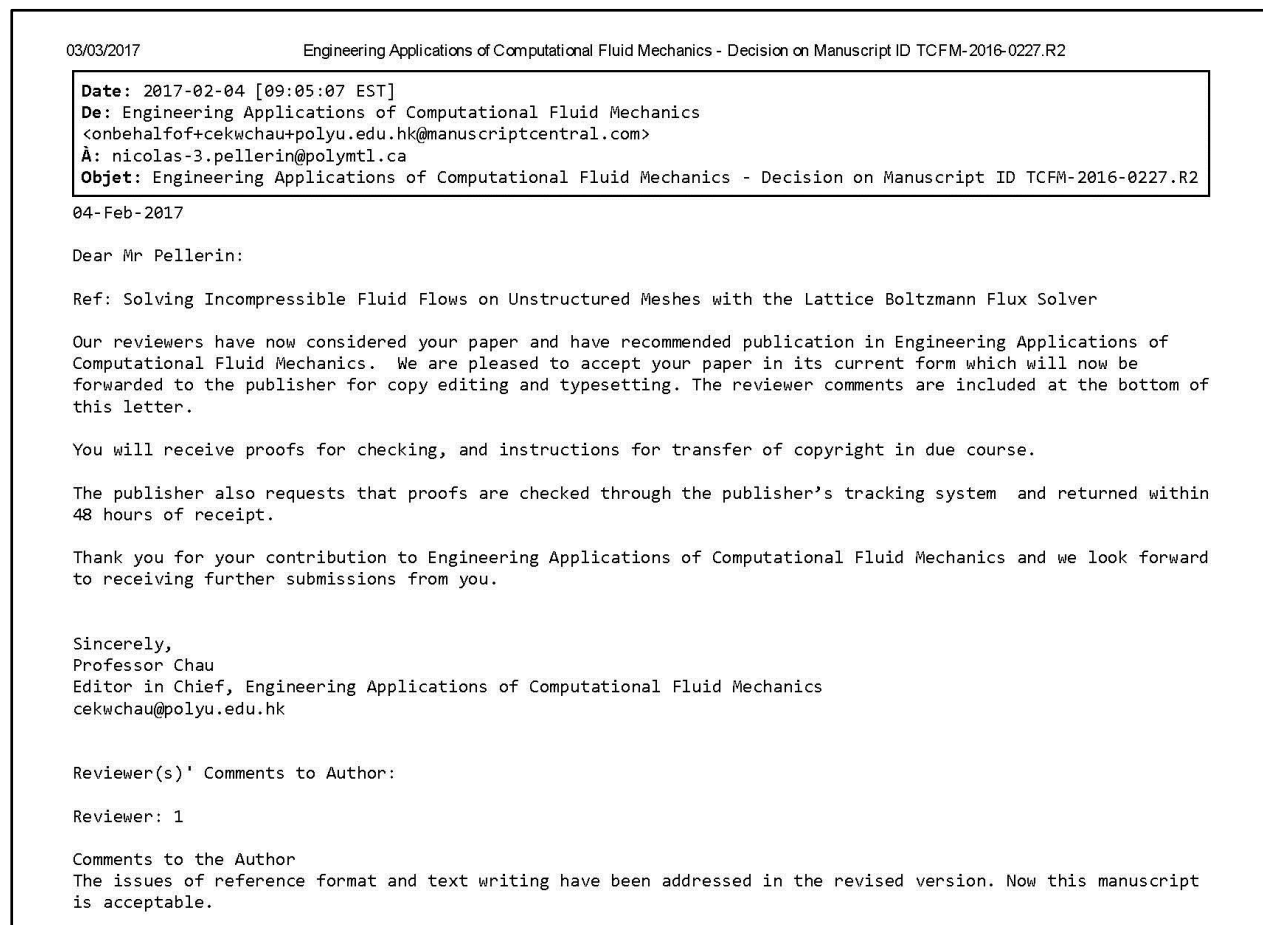


Figure D.1 : Avis d'acceptation par les éditeurs de l'article «Solving incompressible fluid flows on unstructured meshes with the lattice Boltzmann method».

Solving Incompressible Fluid Flows on Unstructured Meshes with the Lattice Boltzmann Flux Solver

Nicolas Pellerin (corresponding)

Department of Mechanical Engineering, Polytechnique Montréal, Montréal, Canada

2500 chemin de Polytechnique, Montréal, Québec H3T 1J4, Canada

nicolas-3.pellerin@polymtl.ca

(613) 297-6176

Sébastien Leclaire

Department of Computer Science, University of Geneva, Geneva, Switzerland

7 route de Drize, 1227 Carouge, Switzerland

sebastien.leclaire@etu.unige.ch

0223790044

Marcelo Reggio

Department of Mechanical Engineering, Polytechnique Montréal, Montréal, Canada

2500 chemin de Polytechnique, Montréal, Québec H3T 1J4, Canada

marcelo.reggio@polymtl.ca

(514) 340-4711, ext. 4411

Solving Incompressible Fluid Flows on Unstructured Meshes with the Lattice Boltzmann Flux Solver

An application of the recently developed lattice Boltzmann flux solver (LBFS) is proposed to solve incompressible flows using unstructured meshes with high aspect ratio triangular cells. The capability to solve turbulent flows is also introduced by coupling the method with a turbulence model, for which the viscosity transport equation is solved on the same mesh. The proposed computational approach is validated for the classical lid-driven cavity flow, the flow over a circular cylinder, and the turbulent flow around a NACA0012 airfoil. Overall, the results obtained agree well with reference data, and demonstrate the validity of using the LBFS on directionally refined meshes, providing the advantage of limiting the number of vertices required in boundary layer regions of the fluid flow. An alternate flux construction method derived from a lattice Boltzmann boundary condition model based on equilibrium distribution streaming is also presented.

Keywords: Lattice Boltzmann method; flux solver; turbulence; Spalart-Allmaras; cylinder; airfoil

Subject classification codes: 76M28, 76M12

Introduction

The lattice Boltzmann method (LBM) for fluids is a flow-solving approach which treats fluids as a probabilistic ensemble of interacting particles, rather than as a continuum, as in Navier-Stokes solvers. This method has become an alternative to established computational strategies for simulating incompressible aerodynamic flows. In fact, commercial packages that can simulate unsteady 3D aerodynamics are available (Holman, Brionnaud, & Modena, 2014; König, Fares, & Nölting, 2014). Research has also been conducted in an effort to simulate turbulent airfoil flows with 2D lattices or meshes (Filippova et al., 2001; Imamura, Suzuki, Nakamura, & Yoshida, 2005; Li, Shock, Zhang, Chen, & Shih, 2005; Li, Zhong, Zhuo, & Cao, 2012; Pellerin, Leclaire, & Reggio, 2015).

For flows with high Reynolds numbers, in a turbulent regime, the LBM needs to be supplemented with turbulence models. The 3D commercial packages referenced above employ large eddy simulation strategies. For 2D flows, the LBM can be coupled with a turbulent viscosity transport equation solver. Regardless of the approach, fast flows require mesh spacing near walls to be small enough to capture the fluid dynamics within the boundary layers. For this, both commercial packages use regular lattices and multi-domain mesh refinement strategies in order to provide adequate lattice spacing near walls, while limiting the total number of lattice sites. In contrast, the approaches of Imamura et al. (2005) and Li, Zhong et al. (2012) are not based on step changes in grid size. Imamura et al. (2005) use an orthogonal grid system that conforms to the airfoil geometry, in conjunction with coordinate transformation that allows the LBM to be applied on a non-Cartesian mesh. Li, Zhong et al. (2012) also use coordinate transformation, however they retain a Cartesian mesh and stretch it to reduce grid spacing in the vicinity of the airfoil. In previous work (Pellerin et al., 2015), the current authors elected to use the multi-domain approach, and coupled it to a finite-difference implementation of the Spalart-Allmaras turbulence model. They

identified mesh refinement near the airfoil's leading edge as a key element in obtaining accurate solutions.

While multi-domain lattices are easy to construct and can be adapted to arbitrary shapes, they do present certain drawbacks. First, their implementation in a solver, such as Pellerin et al.'s (2015) recursive solver, requires expending significant computational effort to transfer information between domains. Second, by definition, regular lattices cannot be refined along one specific direction, which would certainly be an advantage in terms of capturing high velocity gradients normal to walls, while limiting the number of chordwise lattice sites along an airfoil. The method of Imamura et al. (2005) based on a non-uniform orthogonal mesh addresses this issue somewhat, but would not be an optimal solution if the shape of the object subjected to the flow were less regular than a clean airfoil, for instance an airfoil with its leading edge covered with irregular ice shapes. In light of these drawbacks, the current work focuses on using the LBM on unstructured meshes, which allows the solver implementation to be simplified, as well as providing a mesh that can be adapted to any object shape and refined more in the direction normal to a wall than along a wall.

The first unstructured LBM approach was introduced by Peng, Xi, Duncan, and Chou (1998; 1999), who transformed the lattice Boltzmann equation into a vertex-centered finite-volume formulation, and calculated the particle distribution fluxes through the boundaries of the control volumes with linear interpolations. Noting that the finite-volume LBM of Peng et al. (1999) suffers from more numerical instability than the standard LBM, Stiebler, Tölke, and Krafczyk (2006) proposed using an upwind scheme for the discretization of the convection operator. They compared their method to the central discretization scheme of Peng et al. (1999) and achieved stability at higher Reynolds numbers. Observing that central schemes can introduce non-physical oscillations, and that vertex-centered finite-volume formulations require large amounts of memory storage for the boundary fluxes, Patil and Lakshmisha (2009) introduced a cell-centered finite-volume formulation coupled to a Total Variation Diminishing scheme for the flux calculations. They concluded that their method showed better numerical stability than the central scheme. Zarghami, Maghrebi, Ghasemi, and Ubertini (2012) also use a cell-centered formulation, and address the stability issue by introducing upwind second order pressure-based biasing factors and artificial damping in the flux calculations.

More recently, Shu, Wang, Teo, and Wu (2014) introduced the lattice Boltzmann flux solver (LBFS). This method is based on the correspondence between the LBM and the Navier-Stokes equations, which can be demonstrated through a Chapman-Enskog expansion. Unlike the finite-volume LBM, for which particle distribution fluxes are evaluated and used in distribution conservation equations, the LBFS uses particle distributions to compute mass and momentum fluxes, and then updates macroscopic conservation equations. Shu et al. (2014) validated the LBFS with orthogonal meshes on standard test cases, such as the lid-driven cavity flow and the uniform flow past a circular cylinder, obtaining results that compared favorably with those available in the literature. One interesting aspect of the LBFS is the method used to obtain the macroscopic properties necessary for the flux calculations. Shu et al. (2014) use density and velocity gradients at the cell centers and project them onto the locations near the cell boundaries where the values are required to calculate the usual LBM equilibrium distributions necessary for calculating the fluxes. The method for obtaining the gradients is not specified by these authors, nor do they state that an upwind interpolation scheme is required. Even so, they report a stable solution for the lid-driven

cavity flow with a Reynolds number of 10000, using a non-uniform orthogonal grid of 121x121 with near-wall refinement. This suggests that the LBFS may have interesting stability properties that could be useful for higher Reynolds number flows. Furthermore, the usual collide-and-stream process of the standard LBM is by nature an upwind operator in the lattice speed directions, and the LBFS, by the way it is constructed, takes advantage of this.

The LBFS was subsequently used by Wang, Shu, Teo, and Wu (2015) for studying fluid-structure interactions through the incorporation of immersed boundary conditions. They tested their method on flows with stationary and moving circular cylinders and obtained satisfactory results. The method was further extended to 3D flows by Wang, Shu, Teo, and Yang (2016) and conclusively tested on unsteady flows around a sphere and a torus. Even though only non-uniform orthogonal meshes were used in the works referenced, the LBFS is also applicable to unstructured meshes, because of the way it is constructed. In fact, Wang, Yang, and Shu (2015) extended the applicability of the LBFS to compressible flows and used a Delaunay-type triangular unstructured mesh to test their model on a low Reynolds number flow around two airfoils placed in a biplane configuration. This method is promising in terms of stability, as no special care is given to the interpolations used for the flux calculations. This is not the case for the finite-volume LBM, which is why the LBFS was chosen for the current work. Its applicability will be tested on weakly compressible higher Reynolds flows around an airfoil, using high aspect ratio triangles near the wall to provide for differing mesh refinement in the normal and tangential directions. For the turbulence modeling, a finite-difference implementation of the Spalart-Allmaras model is applied, following the work of Pellerin et al. (2015).

We use three validation cases to verify the accuracy of the proposed unstructured LBFS implementation. We first solve the laminar flows in a lid-driven cavity and over a circular cylinder to confirm that this LBFS implementation is accurate when no turbulence modeling is used. We then solve turbulent flows around the NACA0012 airfoil at moderate Reynolds numbers for the same test case as in Pellerin et al. (2015). This provides a basis for comparing multi-domain and unstructured LBM solvers.

This article is organized as follows. In section 2, we provide a description of the LBFS and a vertex-centered application on an unstructured mesh. In section 3, we detail the finite-difference application of the Spalart-Allmaras turbulence model. In section 4, we define the boundary conditions implementation and describe a convergence acceleration method. In sections 5 to 7, we present the results for the lid-driven cavity flow, the circular cylinder flow, and the NACA0012 airfoil flow respectively. In the remaining sections, we discuss our results and present our concluding remarks. In the appendix, we describe an alternative flux construction method based on earlier work by Pellerin, Leclaire, and Reggio (2014) and compare it to Shu et al.'s (2014) method.

2. Unstructured LBFS

The lattice Boltzmann method is based on the Boltzmann equation derived from the kinetic theory of gases (Viggen, 2014):

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f = \Omega(f) , \quad (1)$$

represented here without a body force term. In this equation, $f = f(\mathbf{x}, \mathbf{v}, t)$ represents a particle distribution function at a given location \mathbf{x} and a given time t for particles moving at a microscopic velocity \mathbf{v} . The LHS of Eq. (1) represents convective motion and the RHS the scattering effect of particle collisions. A simple model for the collision process was introduced by Bhatnagar, Gross, and Krook (1954), which consists of a linear relaxation towards a local equilibrium:

$$\Omega(f) = \frac{1}{\tau} (f^{eq} - f), \quad (2)$$

in which the Maxwell-Boltzmann distribution is used for the equilibrium distribution f^{eq} . In order to obtain the lattice Boltzmann equation, the continuous Boltzmann equation must be discretized in time, in physical space, and in velocity space, since the distribution functions depend on these three parameters. For the velocity space discretization, velocity vectors pointing towards the neighboring sites of a regular lattice are introduced, which leads to the discrete velocity Boltzmann equation:

$$\frac{\partial f_i}{\partial t} + \mathbf{c}_i \cdot \nabla f_i = \frac{1}{\tau} (f_i^{eq} - f_i), \quad (3)$$

where \mathbf{c}_i is a given velocity vector along the direction index i . This index covers all the discrete directions according to the chosen lattice configuration. The time and space discretizations are performed by expressing the LHS of Eq. (3) with a material derivative and approximating it with a finite time step. This leads to the lattice Boltzmann equation:

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + \frac{\Delta t}{\Delta t_R} (f_i^{eq} - f_i). \quad (4)$$

In Eq. (4), the commonly used relaxation time symbol τ was replaced with Δt_R to highlight the fact that the particle distribution relaxation coefficient is a ratio of finite time steps, and to show that Δt in lattice units is not necessarily set to unity. Given a velocity space discretization, the equilibrium distributions are obtained by applying a second order Taylor series expansion centered on the macroscopic velocity origin to the Maxwell-Boltzmann distribution (He & Luo, 1997):

$$f_i^{eq}(\mathbf{x}, t) = \rho w_i \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{1}{2} \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{c_s^4} - \frac{1}{2} \frac{\mathbf{u} \cdot \mathbf{u}}{c_s^2} \right), \quad (5)$$

where ρ and \mathbf{u} are the macroscopic density and velocity respectively, and c_s is the lattice speed of sound. The chosen weights w_i ensure that

$$\rho = \sum_{i=1}^N f_i^{eq} = \sum_{i=1}^N f_i, \quad (6)$$

and

$$\rho \mathbf{u} = \sum_{i=1}^N \mathbf{c}_i f_i^{eq} = \sum_{i=1}^N \mathbf{c}_i f_i. \quad (7)$$

For two-dimensional simulations, the d2q9 lattice model is widely used. Its nine velocity vectors are given by

$$\mathbf{c}_i = (c_x, c_y)_i = \begin{cases} (0, 0) & i = 1 \\ c \left(\cos[(i-2)\pi/2], \sin[(i-2)\pi/2] \right) & i = 2, 3, 4, 5 \\ c \left(\cos[(i-6)\pi/2 + \pi/4], \sin[(i-6)\pi/2 + \pi/4] \right) & i = 6, 7, 8, 9 \end{cases} \quad (8)$$

and the associated weights are

$$w_i = \begin{cases} 4/9 & i = 1 \\ 1/9 & i = 2, 3, 4, 5 \\ 1/36 & i = 6, 7, 8, 9 \end{cases} \quad (9)$$

The lattice speed c is defined as $\Delta x / \Delta t$ and is usually set to unity. It is related to the lattice speed of sound through $c_s^2 = c^2 / 3$. Eqs. (4)-(9) constitute the basis of the lattice Boltzmann method termed LBGK. Through a Chapman-Enskog expansion, it can be demonstrated that this model is equivalent to the Navier-Stokes equations applied to weakly compressible flows (Chen & Doolen, 1998; Shu et al., 2014), and that the link between the macroscopic scale of the Navier-Stokes equations (Eq. (10)) and the mesoscopic scale of the LBGK is provided by Eqs. (11)-(12).

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}^T) = -\nabla \cdot (p \mathbf{I}) - \nu (\nabla (\nabla \cdot (\rho \mathbf{u})) + (\nabla \cdot \nabla)(\rho \mathbf{u})) \quad (10)$$

$$p = \rho c_s^2 \quad (11)$$

$$\nu = c_s^2 \Delta t \left(\frac{\Delta t_R}{\Delta t} - \frac{1}{2} \right) \quad (12)$$

In Eq. (12), ν represents the kinematic viscosity in lattice units. Its value is obtained from the Reynolds number of the simulation and the chosen reference speed, also in lattice units.

While the standard approach to solving the flow with the LBGK is to separate Eq. (4) into a collision step and a streaming step, the LBFS of Shu et al. (2014) takes a different approach, based on additional insights obtained from the Chapman-Enskog expansion. In fact, the momentum flux tensor appearing in the Navier-Stokes equations expressed in the following form:

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\Pi) = 0, \quad (13)$$

is equivalent to the following moment of the distribution functions

$$\Pi = \sum_{i=1}^N \mathbf{c}_i \mathbf{c}_i^T \left[f_i^{eq} + \left(1 - \frac{\Delta t}{2\Delta t_R} \right) f_i^{neq} \right]. \quad (14)$$

Also, the non-equilibrium distributions f^{neq} are related to the equilibrium distributions through

$$f_i^{neq} = -\Delta t_R \left(\frac{\partial}{\partial t} + \mathbf{c}_i \cdot \nabla \right) f_i^{eq}. \quad (15)$$

The LBFS uses a finite-volume formulation to solve the continuity and Navier-Stokes equations, and Eqs. (5), (6), (7), (14), and (15) to calculate the cell boundary fluxes. Eq. (15) is not

actually used in its continuous derivative form, but is recast into a backwards time finite-difference equation. To do so, the RHS of the equation is expressed as a material derivative and then transformed into:

$$f_i^{neq}(\mathbf{x}, t) = -\Delta t_R \frac{f_i^{eq}(\mathbf{x}, t) - f_i^{eq}(\mathbf{x} - \mathbf{c}_i \Delta t, t - \Delta t)}{\Delta t} . \quad (16)$$

For a two-dimensional problem, Shu et al. (2014) provide the general finite-volume equations:

$$\mathbf{W} = \begin{Bmatrix} \rho \\ \rho u_x \\ \rho u_y \end{Bmatrix}, \quad \mathbf{E} = \begin{Bmatrix} \rho u_x \\ \Pi_{xx} \\ \Pi_{yx} \end{Bmatrix}, \quad \mathbf{F} = \begin{Bmatrix} \rho u_y \\ \Pi_{xy} \\ \Pi_{yy} \end{Bmatrix}, \quad (17)$$

$$\frac{d\mathbf{W}_j}{dt} = -\frac{1}{dV_j} \sum_{k=1}^K (n_x \mathbf{E} + n_y \mathbf{F})_k dS_k, \quad (18)$$

where dV_j is the control volume, and dS_k is the k^{th} control surface. Since the computations are performed in 2D, unit depth is assumed for the control volume and surfaces. In the original presentation of the LBFS (Shu et al., 2014), and in following work (Wang, Shu, Teo & Wu, 2016; Wang, Shu, Teo & Yang, 2015), a cell-centered approach is taken. The present implementation differs in that a vertex-centered scheme for the finite-volume computations is used. This method was selected because the macroscopic quantity interpolations performed to compute the boundary fluxes can be simply accomplished with linear interpolations, whereas in the cell-centered approach the quantities are obtained with projections based on spatial gradient approximations. Furthermore, the vertex-centered approach provides more points for discretizing a control volume boundary, which is advantageous when high aspect ratio triangular cells are used, because the fluxes in the elongated direction can be better approximated. In fact, some care is generally required when using slender cells, and the aspect ratio should be limited when high gradients are expected in all directions. In the present work, wake recirculation is expected for the cylinder simulations, therefore the aspect ratio will be limited in order to accurately capture the separation angle. On the other hand, the NACA 0012 airfoil flows are expected to remain attached, therefore higher aspect ratios will be used on the mid-chord wall section, where the gradients are much larger in the wall normal direction than in the tangential direction.

A typical unstructured mesh vertex-centered control volume is represented in Fig. 1a. Each triangle bounding the vertex provides two surfaces through which fluxes may pass, calculated at points C and D shown in Fig. 1b. These points are placed halfway between the triangle's geometric center and its side midpoints. Additional surfaces can be included if the triangle is bounded by the domain limit or an object wall, as indicated by the G and H flux calculation points. For adjacent triangles, both of which are located in the flow domain, the fluxes calculated at these points are internal to the control volumes and cancel out, and therefore are omitted.

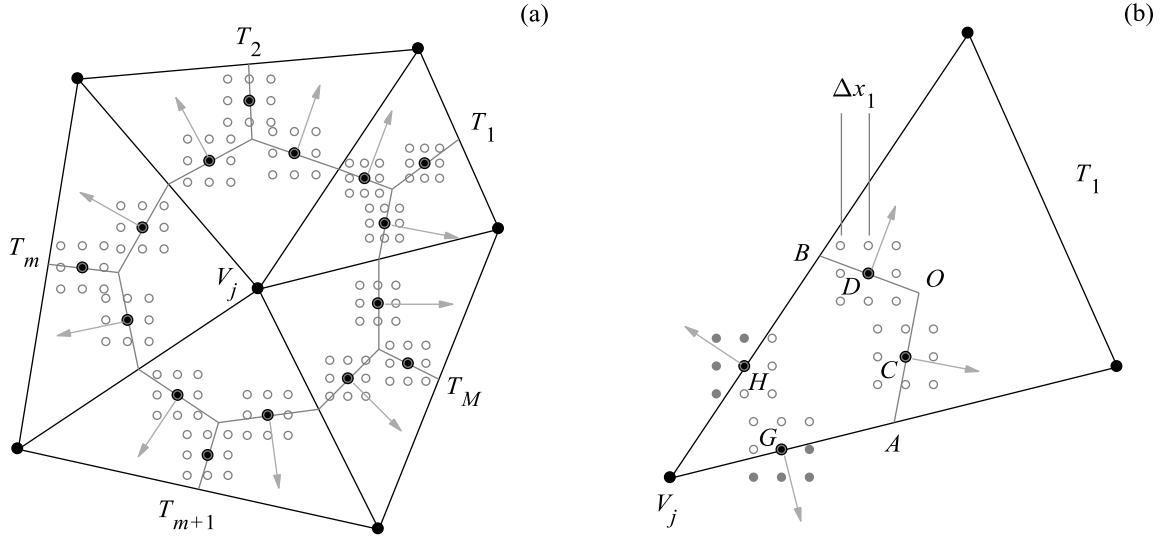


Figure 1. a) Unstructured mesh vertex-centered control volume schematic, and b) triangular cell with depiction of core and boundary flux calculation points.

For each flux calculation point, nine macroscopic interpolation points are included, as depicted in Fig. 1. These latter points correspond to the mirrored d2q9 lattice configuration and are used for constructing particle distributions, which will be streamed onto the flux calculation points. Each interpolation point is located at $\mathbf{x}_{flux} - \Delta \mathbf{x}_i = \mathbf{x}_{flux} - \mathbf{c}_i \Delta t$, which is, in fact, equivalent to $\mathbf{x}_{flux} - \mathbf{c}_i \Delta x$, since we assume a unitary lattice speed $c = \Delta x / \Delta t = 1$. The Δx value is chosen to be identical for all flux calculation points within a particular triangle, and is set to ensure that all nine macroscopic interpolation points linked to the core flux calculation points, for instance C or D , are within the triangle. Since the interpolations are to be performed linearly, the rationale behind this choice for the interpolation point locations is to use only local triangle information in order to ensure local continuity of the macroscopic fields, while avoiding extrapolations outside the triangle. Obviously, for boundary points, such as G or H , some macroscopic interpolation points will lie outside the triangle, and so their location will be approximated to that of the center point of the d2q9 stencil.

Within the entire domain, each flux calculation point has its own Δt_R , which is calculated via Eq. (12) and again using $\Delta t = \Delta x$, or $c = 1$.

The procedure for calculating the fluxes and updating the macroscopic quantities within a control volume is described below. It is directly inspired by the work of Shu et al. (2014), however a different macroscopic time stepping method is used.

- (1) Interpolate ρ and \mathbf{u} linearly on the macroscopic interpolation points, using the values at the vertices of the local triangle.
- (2) Use the interpolated macroscopic quantities in Eq. (5) to construct equilibrium distributions.

- (3) Stream the equilibrium distributions obtained in step 2 onto the associated flux calculation points, and compute ρ_{flux} and \mathbf{u}_{flux} using Eqs. (6)-(7).
- (4) Calculate the equilibrium distributions at the flux calculation points using Eq. (5), and ρ_{flux} and \mathbf{u}_{flux} obtained in step 3.
- (5) Calculate the non-equilibrium distributions at the flux calculation points using Eq. (16), in which the equilibrium distributions $f_i^{eq}(\mathbf{x} - \mathbf{c}_i \Delta t, t - \Delta t)$ and $f_i^{eq}(\mathbf{x}, t)$ come from steps 2 and 4 respectively.
- (6) Compute the components of the momentum flux tensor Π using the equilibrium and non-equilibrium distributions of steps 4 and 5, and Eq. (14).
- (7) Build the \mathbf{E} and \mathbf{F} vectors of Eq. (17) with ρ_{flux} , \mathbf{u}_{flux} , and Π .
- (8) Update \mathbf{W} , that is, the vertex ρ and \mathbf{u} , using Eq. (18) and the Euler forward method for time stepping.

The magnitude of the time step used in step 8 is not the same as the time step associated with the flux calculation points. In order for the time evolution of the solution in every control volume to follow the same transient time scale, this time step must be identical for all control volumes. It is set to the Δt of the smallest triangle (or Δx since c is set to 1) in the entire domain. This condition is later relaxed to increase the convergence rate for assumed steady flows. Shu et al. (2014) use a fourth-order Runge-Kutta scheme for time integration, but they do not specify the value they use for Δt . While a higher order integration scheme is expected to be more accurate, for the present LBFS implementation, a first-order scheme was selected for its simplicity and because we consider the chosen time step to be small enough for the integration scheme to retain sufficient accuracy.

3. Spalart-Allmaras Turbulence Model

For the third test case, we simulate turbulent airfoil flows. Since these simulations are two-dimensional, we need to use a turbulence model of the Reynolds Averaged Navier-Stokes type, for which all turbulent scales are modeled. Based on previous work by Pellerin et al. (2015), we chose the Edwards version of the Spalart-Allmaras model. The model equations are not repeated here. The current work also uses a finite-difference scheme. However, since the present method uses an unstructured mesh, the velocity and viscosity gradient calculation stencil is different and should be detailed. Therefore, Fig. 2 displays the interpolation points N , S , E , and W used to compute the gradients at a given vertex. The interpolations are linear and use the vertices at each end of the local triangle edge.

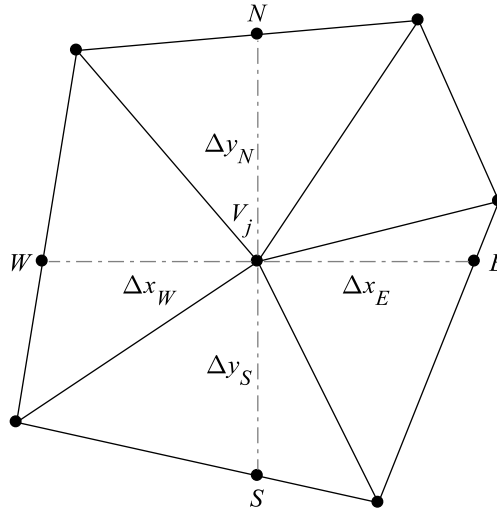


Figure 2. Gradients calculation stencil.

The first-order gradients of the Spalart-Allmaras viscosity $\tilde{\nu}$ are calculated with an upwind scheme:

$$\frac{\partial \tilde{\nu}}{\partial x} = \begin{cases} (\tilde{\nu}_V - \tilde{\nu}_W) / \Delta x_W & u_{x,V} \geq 0 \\ (\tilde{\nu}_E - \tilde{\nu}_V) / \Delta x_E & u_{x,V} < 0 \end{cases}, \quad (19)$$

$$\frac{\partial \tilde{\nu}}{\partial y} = \begin{cases} (\tilde{\nu}_V - \tilde{\nu}_S) / \Delta y_S & u_{y,V} \geq 0 \\ (\tilde{\nu}_N - \tilde{\nu}_V) / \Delta y_N & u_{y,V} < 0 \end{cases}, \quad (20)$$

and the second-order gradients are calculated with a central-difference scheme:

$$\frac{\partial^2 \tilde{\nu}}{\partial x^2} = \frac{(\tilde{\nu}_E - \tilde{\nu}_V) / \Delta x_E - (\tilde{\nu}_V - \tilde{\nu}_W) / \Delta x_W}{0.5(\Delta x_E + \Delta x_W)}, \quad (21)$$

$$\frac{\partial^2 \tilde{\nu}}{\partial y^2} = \frac{(\tilde{\nu}_N - \tilde{\nu}_V) / \Delta y_N - (\tilde{\nu}_V - \tilde{\nu}_S) / \Delta y_S}{0.5(\Delta y_N + \Delta y_S)}. \quad (22)$$

The velocity component gradients used for calculating \tilde{S} in the viscosity production term are calculated with a central-difference scheme:

$$\frac{\partial u_x}{\partial x} = \frac{u_{x,E} - u_{x,W}}{\Delta x_E + \Delta x_W}, \quad \frac{\partial u_y}{\partial x} = \frac{u_{y,E} - u_{y,W}}{\Delta x_E + \Delta x_W}, \quad (23)$$

$$\frac{\partial u_x}{\partial y} = \frac{u_{x,N} - u_{x,S}}{\Delta y_N + \Delta y_S}, \quad \frac{\partial u_y}{\partial y} = \frac{u_{y,N} - u_{y,S}}{\Delta y_N + \Delta y_S}. \quad (24)$$

The time step used for the forward time-stepping of the Spalart-Allmaras equation is the same as for the finite-volume method in the LBFS. A turbulent viscosity ν_t is obtained at every vertex from the updated $\tilde{\nu}$. It is then interpolated linearly at the flux calculation points, and the

relaxation times Δt_R are updated with Eq. (12) by replacing ν with $\nu + \nu_t$. On solid walls, the turbulent viscosity should be null, and so $\tilde{\nu}$ is set to 0 on the associated vertices.

Some explanation should be provided for the choice of a finite-difference scheme for solving the Spalart-Allmaras equation. Since a finite-volume scheme is adopted for solving the flow, it would have been logical to apply a similar scheme for the turbulence transport equation. However, the current approach was chosen to limit memory usage. In this vertex-centered finite-volume implementation, each vertex can have up to 10 bounding triangles, depending on the local mesh topology. This may result in as many as 20 flux points to compute, which would have a significant impact on memory requirements. The proposed finite-difference scheme requires far less memory. Furthermore, it is also simpler to compute the upwind viscosity gradients with the finite-difference scheme.

4. Boundary Conditions and Convergence Acceleration

The boundary conditions are applied after the macroscopic properties have been updated through Eq. (18) in step 8 of the LBFS procedure detailed above. For the lid-driven cavity flow, the no-slip condition on the walls is applied by setting $\mathbf{u} = (u_x, u_y)$ to $(0, 0)$ for all vertices located on those walls. For the top boundary, \mathbf{u} is set to $(U_0, 0)$ and ρ is set to ρ_0 . For the free stream boundary conditions on a rectangular domain, the velocity is set to $(U_\infty, 0)$ on the left boundary, defined as the inlet. On the top and bottom boundaries, u_y is left free, and an approximate null derivative normal to the boundary is applied to u_x and ρ . This is done by setting the streamwise velocity and density of the boundary vertices to their values interpolated at a distance equivalent to half the local boundary mesh spacing from the boundary. A natural choice for the right boundary, defined as the outlet, would be to impose constant back pressure through the density ρ_0 . However, this approach leads to undesired pressure wave reflections in the computational domain. So instead, a convective condition is imposed for all macroscopic properties, such as

$$\frac{\partial q}{\partial t} + u_x \frac{\partial q}{\partial x} = 0, \quad (25)$$

where q can represent ρ , u_x , or u_y . In Eq. (25), the spatial derivative is calculated with a first-order finite difference using interpolated values from the previous time step. The temporal derivative is calculated with a forward first-order Euler integration. Also, in this implementation of the convective boundary condition, the convective speed u_x is actually set to U_o . Since the density is not strictly imposed on any domain boundary, we ensure that the entire flow field density remains near the nominal ρ_0 level by adding the following quantity to the density of all the vertices:

$$\Delta \rho = \rho_0 - \bar{\rho}_L, \quad (26)$$

where $\bar{\rho}_L$ is the average density on the inlet. This approach is similar to that of Wang, Shu, Teo and Wu (2015), in which the density is left as a free parameter that varies in time. The difference in the present approach is the continuous correction intended to obtain a true steady-state solution of the flow field.

For a flow that is expected to be steady, such as the flow around an airfoil for which the LBFS is supplemented with a turbulence model that treats all turbulence scales in a RANS fashion, the solution convergence can be accelerated using local time-stepping. In this method, the global time step Δt introduced in section 2 is replaced with a local time step Δt_L that is unique for every vertex. It is actually set to the smallest Δt , or Δx , of all the triangles comprised in a vertex's control volume. This means that the physical solution will evolve faster where the mesh cells are larger. Since the cells are much larger far from the airfoil than near its wall, the advantage of this method is to accelerate the influence of the domain boundary conditions on the area near the airfoil. Furthermore, delaying the application of local time-stepping on the continuity equation by a few tens of thousands of iterations also improves convergence time, because it reduces solution oscillations introduced by the initial pressure waves generated at the airfoil wall. The use of a local time-stepping strategy in the context of lattice Boltzmann methods on unstructured meshes is also suggested by Patil and Lakshmisha (2009).

5. Lid-Driven Cavity Flow

The first validation case consists of the classical lid-driven cavity flow, and its purpose is to validate the proposed unstructured LBFS model on a laminar flow and a mesh that does not contain large cell area variations. The flows for Reynolds numbers 100 and 1000 are simulated on three meshes. The coarsest mesh, built with $N=65$ vertices on each of the square domain sides, is shown in Fig. 3. The other two meshes were built with 129 and 257 side vertices respectively. All the meshes were generated with the Gmsh software of Geuzaine and Remacle (2009). The same software was used to build the meshes presented in the sections that follow.

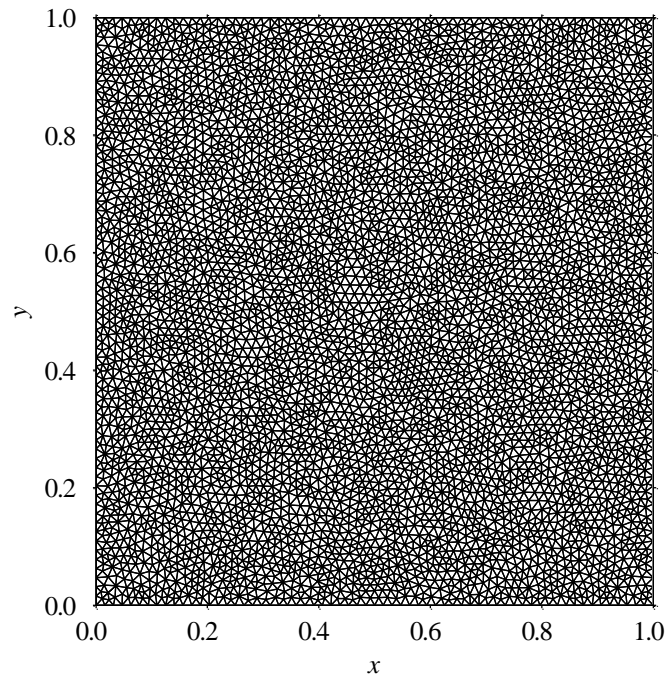


Figure 3. Unstructured mesh for the lid-driven cavity flow, with $N=65$.

The simulations were initialized with the fluid at rest, and calculation was stopped when the normalized momentum sampled every ten thousand time steps did not vary by more than 10^{-5} for any vertex:

$$\max \left(\left| \Delta \left[\rho |\mathbf{u}| / \rho_0 U_0 \right] \right| \right)_{\text{vertices}} \leq 10^{-5} . \quad (27)$$

This convergence criterion was sufficient for reaching steady-state on all meshes. The results are presented in terms of the velocity profiles on domain centerlines, the center location of the main vortex, and the stream function and vorticity magnitude at the vortex center. We compare these results to those of Ghia, Ghia, and Shin (1982), which were obtained by solving a vorticity stream function formulation of the Navier-Stokes equations with a finite-difference method on a 129×129 uniform grid. The normalized velocity profiles for the computations performed with all mesh levels are shown in Figs. 4-5, where

$$x' = -1 + 2x/L_x, \quad y' = -1 + 2y/L_y . \quad (28)$$

The agreement between our results and those of Ghia et al. (1982) is satisfactory for both Reynolds numbers. The velocity profiles for the two finest mesh levels are almost superimposed, which indicates that mesh convergence is nearly attained. Further insight into our solutions is presented in Table 1. For the three meshes used, and for both Reynolds numbers, the locations calculated for the main vortex center agree with the reference data. The stream function and vorticity also agree reasonably well. The results for $Re=1000$ and the $N=129$ meshes are all within 1% of each other.

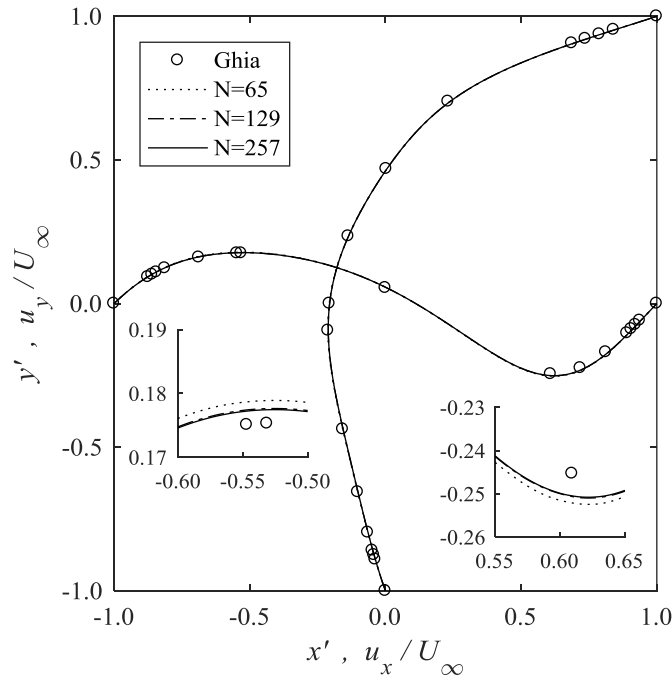


Figure 4. Normalized velocity profile comparison for $Re=100$. The circles represent the results of Ghia et al. and the full line represents the present solution, with $N=257$.

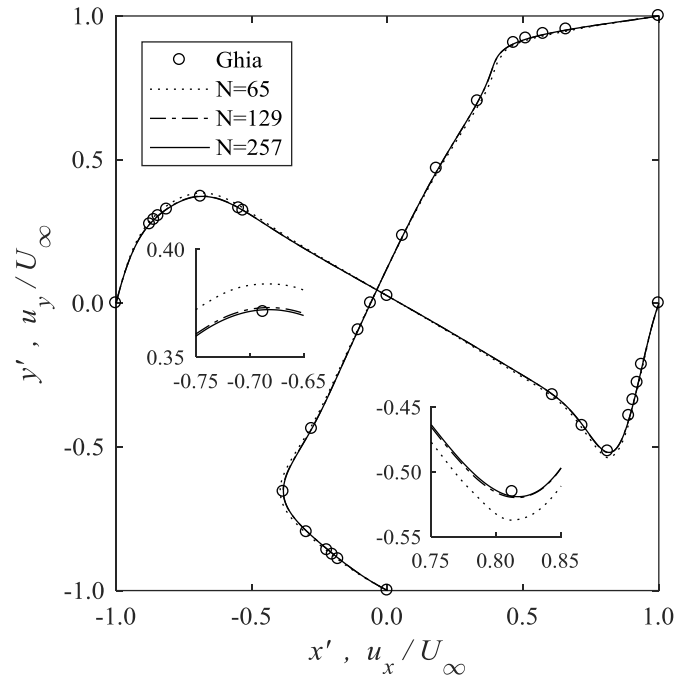


Figure 5. Normalized velocity profile comparison for $Re=1000$. The circles represent the results of Ghia et al. and the full line represents the present solution, with $N=257$.

Re	N	Present				Ghia et al.			
		x_{ctr}	y_{ctr}	ψ	ω	x_{ctr}	y_{ctr}	ψ	ω
100	65	0.6138	0.7359	-0.1032	3.1362				
	129	0.6146	0.7360	-0.1028	3.1270	0.6172	0.7344	-0.103423	3.16646
	257	0.6150	0.7363	-0.1027	3.1345				
1000	65	0.5297	0.5652	-0.1212	2.1102				
	129	0.5305	0.5658	-0.1182	2.0584	0.5313	0.5625	-0.117929	2.04968
	257	0.5306	0.5656	-0.1179	2.0573				

Table 1. Main vortex characteristics for the lid-driven cavity flow.

6. Circular Cylinder in a Free Stream

The second validation case involves the circular cylinder in a free stream flow, and is a case that has been well studied. It was chosen to validate the proposed unstructured LBFS model on a laminar flow and a mesh that contains large cell area variations throughout the computational domain. Cells with larger aspect ratios are also included near the solid wall. The flows for Reynolds numbers ranging between 20 and 200 are simulated on three meshes. The cylinder is located in the center of a square computational domain of side length 200m, and its diameter is set to 1m. For all meshes, 101 vertices are used along each domain side. For the coarsest mesh, represented in Fig. 6, the height of the first vertex layer is set to $h_1 = D/256$, and the distance

between the vertices located on the cylinder wall is constant, at $l_w \approx 3h_1$. The total number of vertices is 60021. The second mesh uses the same aspect ratio for l_w , with $h_1 = D/512$. This yields a total of 63541 vertices. The third mesh also uses the same l_w , and with $h_1 = D/1024$, yielding 70494 vertices.

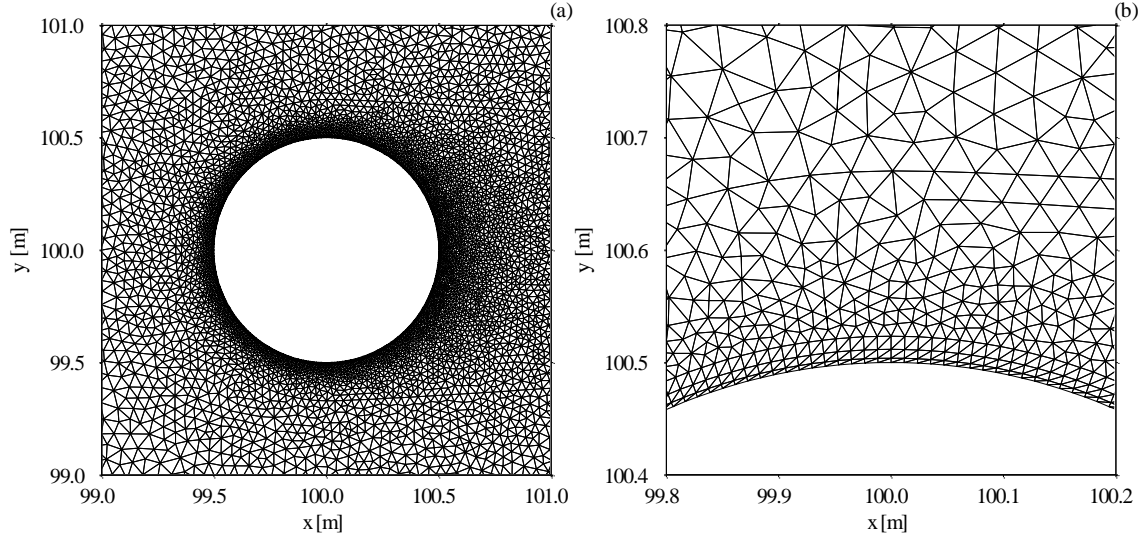


Figure 6. Coarsest unstructured mesh for the circular cylinder flow.

First, in Table 2, we present the drag coefficients, recirculation lengths, and separation angles obtained for the steady flows at Reynolds numbers 20 and 40. We compare them to the results of other works based on a combination of the Boltzmann equation and the finite-volume approach. Zarghami et al. (2012) use a finite-volume LBM formulation on a structured non-uniform mesh. Shu et al. (2014) and Wang, Shu, Teo and Wu (2015) use the LBFS, although the former use a structured curvilinear O-type mesh, and the latter use a structured non-uniform mesh. Yuan, Zhong, and Zhang (2015) use a gas kinetic scheme on a Cartesian mesh with local grid refinement. Their results are included in this comparison because their gas kinetic approach utilizes the BGK (Bhatnagar et al., 1954) collision model and the Maxwellian equilibrium distributions to compute the cell boundary fluxes, and so it is related to the LBFS. More details on this method can be found in Tian, Xu, Chan, and Deng (2007). The results of Sun, Shu, Wang, Teo, and Chen (2016) are presented because they also use a gas kinetic flux solver on a refined Cartesian mesh.

Source	$Re=20$			$Re=40$		
	C_d	l_s / D	θ_s	C_d	l_s / D	θ_s
Zarghami et al.	2.205	0.91	42.5°	1.551	2.235	52.8°
Shu et al.	2.062	0.935	42.94°	1.53	2.240	52.69°
Wang, Shu et al.	2.035	0.93		1.539	2.23	
Yuan et al.	2.071	0.934		1.548	2.287	
Sun et al.	2.07	0.94		1.546	2.36	
Present method, mesh 1	2.003	0.914	43.32°	1.502	2.260	53.29°
Present method, mesh 2	2.007	0.916	43.48°	1.503	2.260	53.52°
Present method, mesh 3	2.010	0.916	43.58°	1.505	2.259	53.64°

Table 2. Drag coefficients, recirculation lengths, and separation angles for the steady flow over a circular cylinder in a free stream flow.

The overall agreement with the selected reference results is good, although our drag coefficients are lower. Also, there is some dispersion in the drag results selected for comparison, which is most likely due to the fact that, for free stream flow problems, domain size, domain boundary conditions, and the mesh selected are all factors that have a large impact on the solution. These parameters were not the same for all the works presented. In the present simulations, it was found that the domain size had a larger impact on the lower Reynolds number flows, and that using smaller domains increased the drag coefficients. This partially explains the lower drag coefficients that were obtained, since our 200Dx200D domain is much larger than that for all the other results presented. In fact, Zarghami et al. (2012) use the smallest domain height, 16D, and obtain a much larger drag coefficient for $Re=20$.

In the present results, grid convergence is not fully attained, and further grid refinement to, perhaps, $h_1 = D/2048$ or $h_1 = D/4096$ would be necessary. Nevertheless, the results are consistent and the drag coefficient difference between successive meshes is small.

Unsteady flow fields were also simulated, at Reynolds numbers of 100 and 200, for which there is regular vortex shedding in the wake of the cylinder. It is therefore important, when comparing results, to include force coefficient variation amplitudes and a non-dimensional measure of the frequency of vortex shedding, namely the Strouhal number:

$$St = \frac{fD}{U_\infty} . \quad (29)$$

Our results are presented in Table 3. Comparison results from the same sources as in Table 2 are included. The subscripts m and a designate the mean and amplitude values respectively.

Source	Re=100				Re=200			
	$C_{d,m}$	$C_{d,a}$	$C_{l,a}$	St	$C_{d,m}$	$C_{d,a}$	$C_{l,a}$	St
Zarghami et al.	1.310			0.161				
Shu et al.	1.334	0.009	0.33	0.164	1.338	0.045	0.69	0.197
Wang, Shu et al.	1.334	0.012	0.37	0.163	1.43	0.051	0.75	0.195
Yuan et al.	1.352			0.164				
Sun et al.	1.367	0.01	0.348	0.164	1.370	0.049	0.714	0.195
Present method, mesh 1	1.325	0.009	0.324	0.164	1.325	0.044	0.673	0.195
Present method, mesh 2	1.325	0.009	0.325	0.164	1.327	0.044	0.677	0.195
Present method, mesh 3	1.326	0.009	0.325	0.164	1.327	0.044	0.677	0.195

Table 3. Force coefficients for the unsteady flow over a circular cylinder in a free stream.

The mean drag coefficients and Strouhal numbers agree very well with the selected reference data. Our results also vary less with mesh level increments than for the lower Reynolds number simulations, therefore grid convergence requires less mesh refinement.

For the coarsest mesh level, simulations for $Re=45$, 47, and 50 were also conducted. The intent was to verify whether or not the transition to unsteady flow near $Re=46$ is captured correctly with our LBFS implementation. The results are displayed in Fig. 7. The flow remains steady for $Re=45$, and is unsteady for $Re=47$, with a Strouhal number of 0.117.

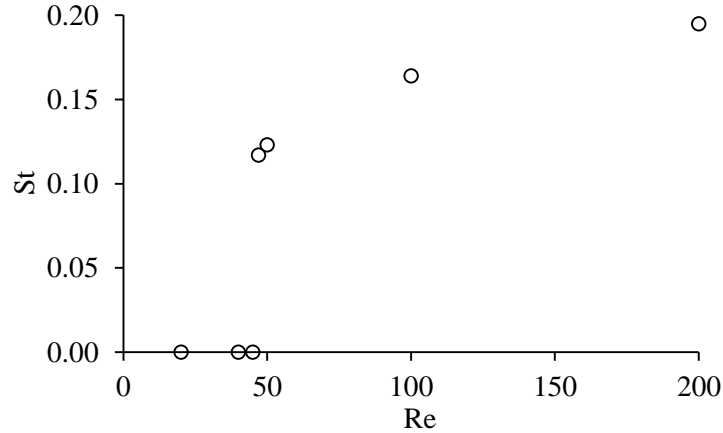


Figure 7. Strouhal number for the flow over a circular cylinder at the coarsest mesh level.

7. NACA0012 Airfoil in a Free Stream

The third validation case involves a turbulent flow over a NACA0012 airfoil at a Reynolds number of 5×10^5 . This case was chosen for two reasons: first, to validate the combining of the LBFS method on an unstructured mesh with a finite-difference implementation of the Spalart-Allmaras turbulence model; and second, to provide a basis for comparing this case with earlier work by the current authors (Pellerin et al., 2015), which is based on a lattice Boltzmann solver applied to a multi-domain lattice. The computational results of Lockard, Luo, Milder, and Singer (2002) obtained with CFL3D are used as the baseline for this comparison.

The computational domain is a 100 m square with 51 vertices along each side. The airfoil is placed in the center of the domain and has a 1-m chord. In contrast with the cylinder meshes, the maximum aspect ratio of the cells near the wall is much larger. Furthermore, only one mesh is used for each of the angles of attack simulated. Although grid refinement necessarily has an impact on the results, the intention was not to study its influence, but rather validate the proposed method. Therefore, the meshes were created with the intention of providing a normal direction refinement near the wall similar to that of the lattices used by Pellerin et al. (2015), and also of requiring fewer vertices along the wall. Actually, the fact that regular lattices cannot be refined more in a given direction was one of the motivations for this work. Simulations were performed for three angles of attack: 0° , 3° , and 7° . The associated meshes were all constructed with two zones. The first zone, near the wall, contains thirty layers of structured triangles. The thickness of the triangles touching the wall is $1/16384^{\text{th}}$ of a chord. The second zone is a fully unstructured mesh joining the structured layers and the domain boundaries. These two zones are depicted in Fig. 8 for the 0° mesh. Along the wall, the distances between vertices is $1/8192^{\text{th}}$ of a chord near the leading edge, and gradually increases to $1/1024^{\text{th}}$ of a chord near the trailing edge. For the 0° mesh, 1001 vertices are located on both the upper and lower surfaces of the airfoil. For the 3° and 7° meshes, 1201 vertices are located on the upper side to provide for more concentration near the leading edge, and 1001 vertices are located on the lower side. The resulting total number of vertices is 131049 for the 0° mesh, 139568 for the 3° mesh, and 139156 for the 7° mesh.

The simulations were run on a desktop computer with a Matlab program implementation of the proposed LBFS method. Local time-stepping was used, and the minimum time step was 6.36×10^{-6} for all the meshes. Time units are not specified because the time steps are in lattice units and stem from the $c = \Delta x / \Delta t = 1$ definition. They do not represent actual physical time, even though they are applied to the macroscopic conservation equations. Convergence of the force coefficients to the fourth decimal place required 2.4×10^5 iterations for the 0° simulation and a total elapsed clock time of 5.6 hours. The 3° simulation required 4.1×10^5 iterations and 11.4 hours, and the 7° simulation 5.3×10^5 iterations and 13.2 hours. The results are presented in terms of drag and lift coefficients, pressure coefficients, and velocity profiles. The force coefficients are provided in Table 4. There is good agreement for lift at 3° and 7° , with a difference of less than 0.7% from the results of the earlier work of Pellerin et al. (2015) and the CFL3D data of Lockard et al. (2002). The drag is in good agreement with the previous work of the current authors, however it still underpredicts the CFL3D results slightly. The pressure coefficients are presented in Figs. 9-10. Good overall agreement with the CFL3D results is observed, with slight differences near the leading edge. The velocity profiles are depicted in Figs. 11-12. The normalized coordinates u^*/U_∞ and y^*/l are defined in Pellerin et al. (2015). There is a close agreement with the authors' earlier work, and the velocity profiles are still fuller than those obtained with CFL3D. Finally, a contour plot of the ratio of turbulent viscosity to nominal fluid viscosity obtained with the present method is provided in Fig. 13 for the 7° angle of attack. This plot provides an indication of the turbulence level in the boundary layer and the near wake of the airfoil. The maximum value of this ratio is approximately 75 and occurs downstream of the trailing edge, after the boundary layers of the pressure and suction sides of the airfoil have merged.

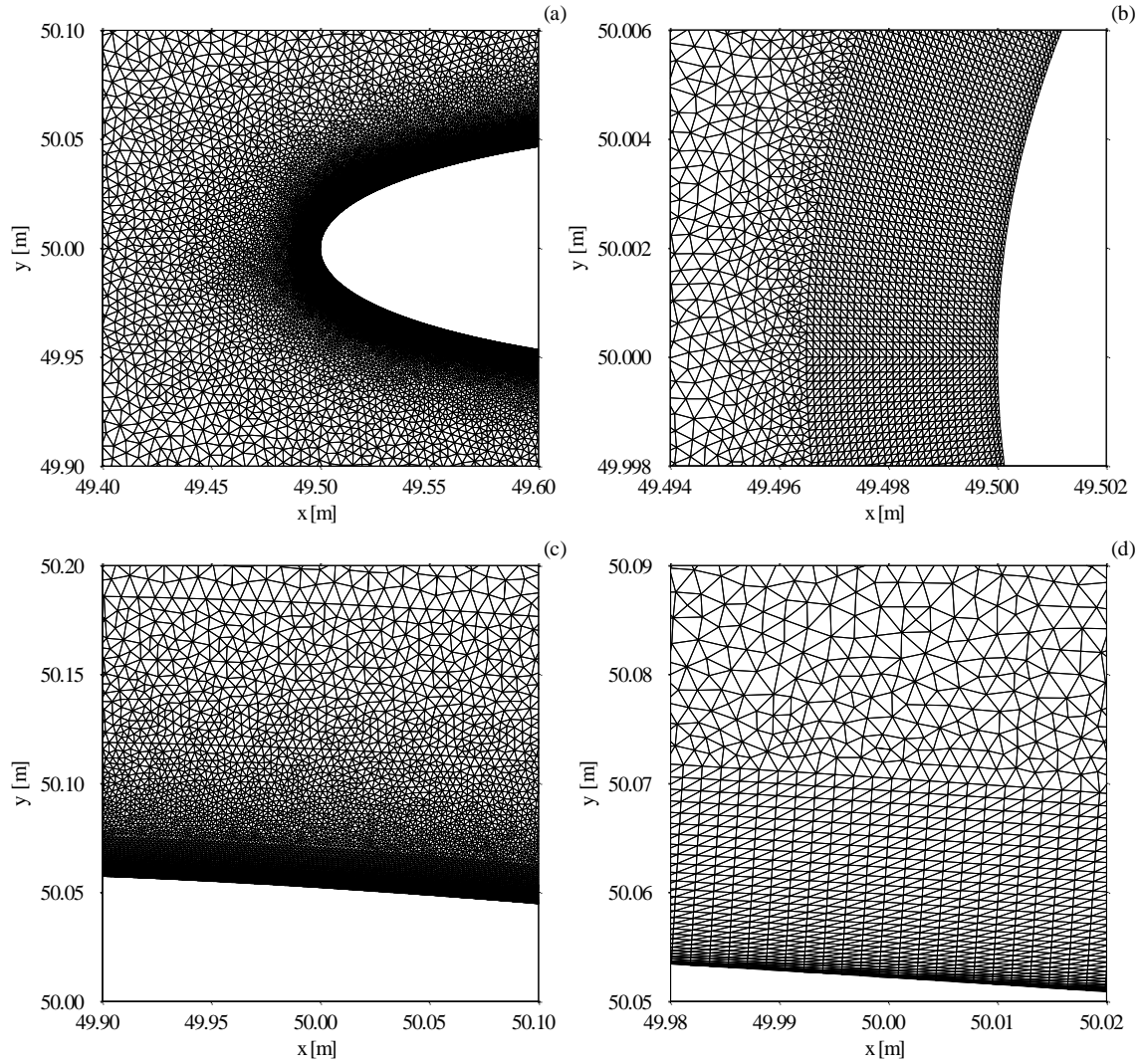


Figure 8. Unstructured mesh for the NACA0012 airfoil at a 0° angle of attack.

AOA [°]	C_L			C_D		
	CFL3D	Pellerin et al.	Present	CFL3D	Pellerin et al.	Present
0	0.0000	0.0000	0.0000	0.0128	0.0100	0.0095
3	0.3237	0.3257	0.3259	0.0130	0.0104	0.0099
7	0.7449	0.7457	0.7418	0.0157	0.0128	0.0122

Table 4. Lift and drag coefficients for the NACA0012 airfoil at $Re=5 \times 10^5$.

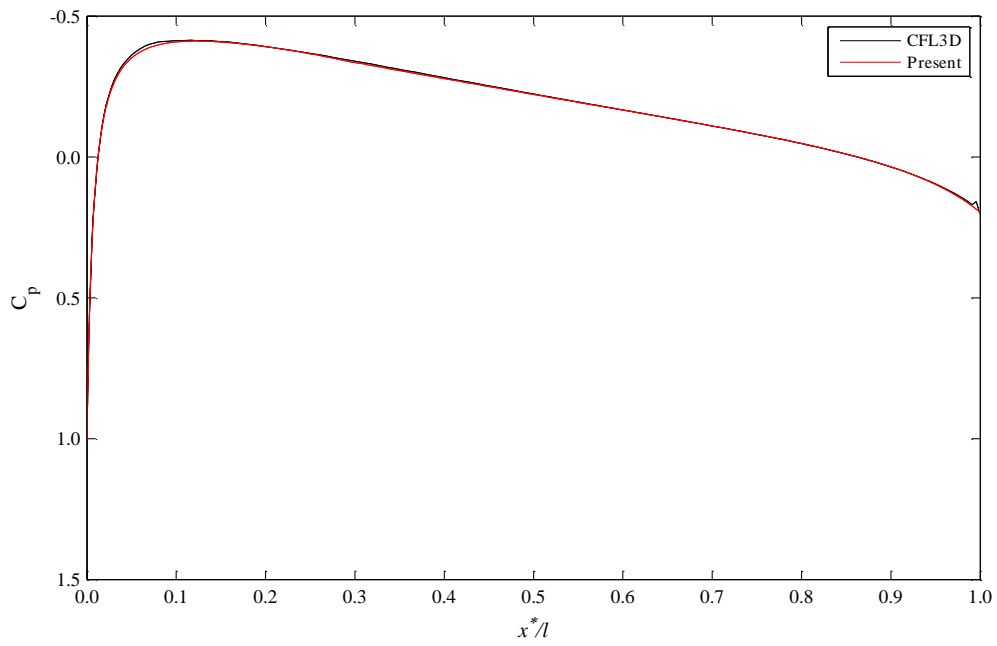


Figure 9. Pressure coefficients for the NACA0012 airfoil at a 0° angle of attack.

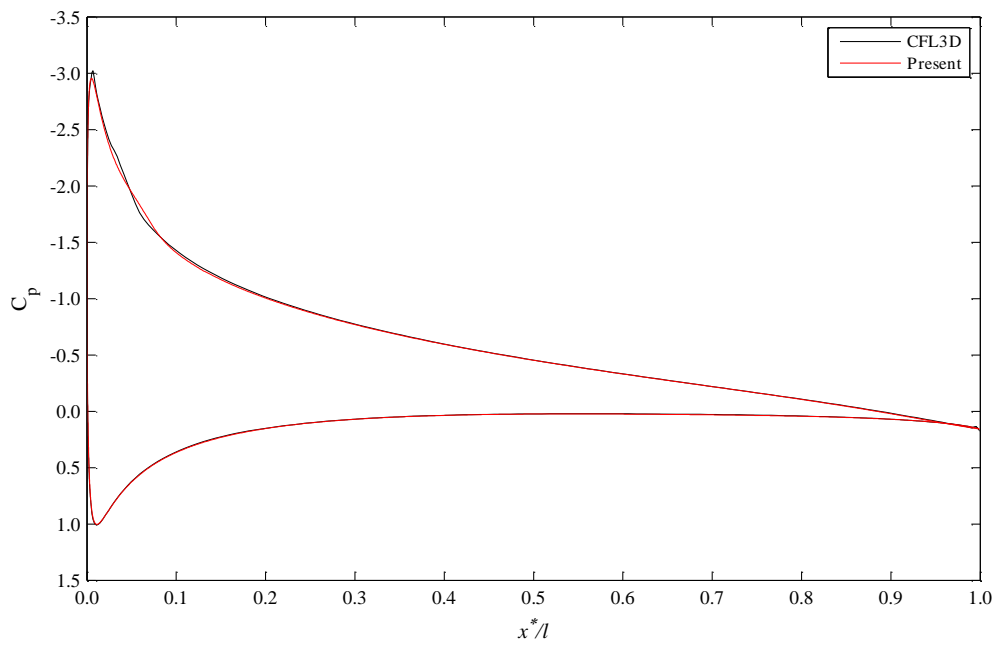


Figure 10. Pressure coefficients for the NACA0012 airfoil at a 7° angle of attack.

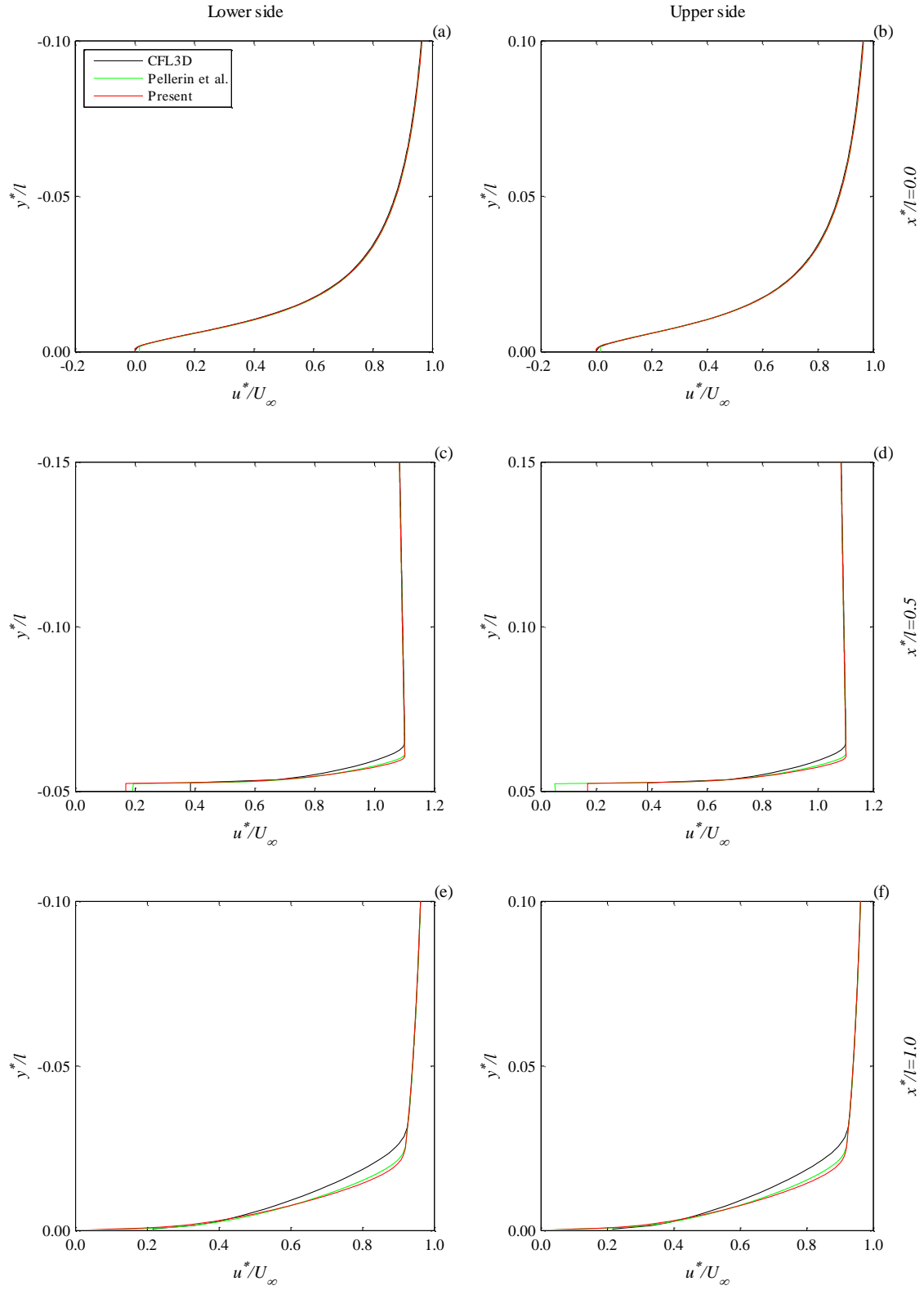


Figure 11. Velocity profiles for the NACA0012 airfoil at a 0° angle of attack at chord coordinates $x^*/l=0.0$, 0.50 , and 1.0 .

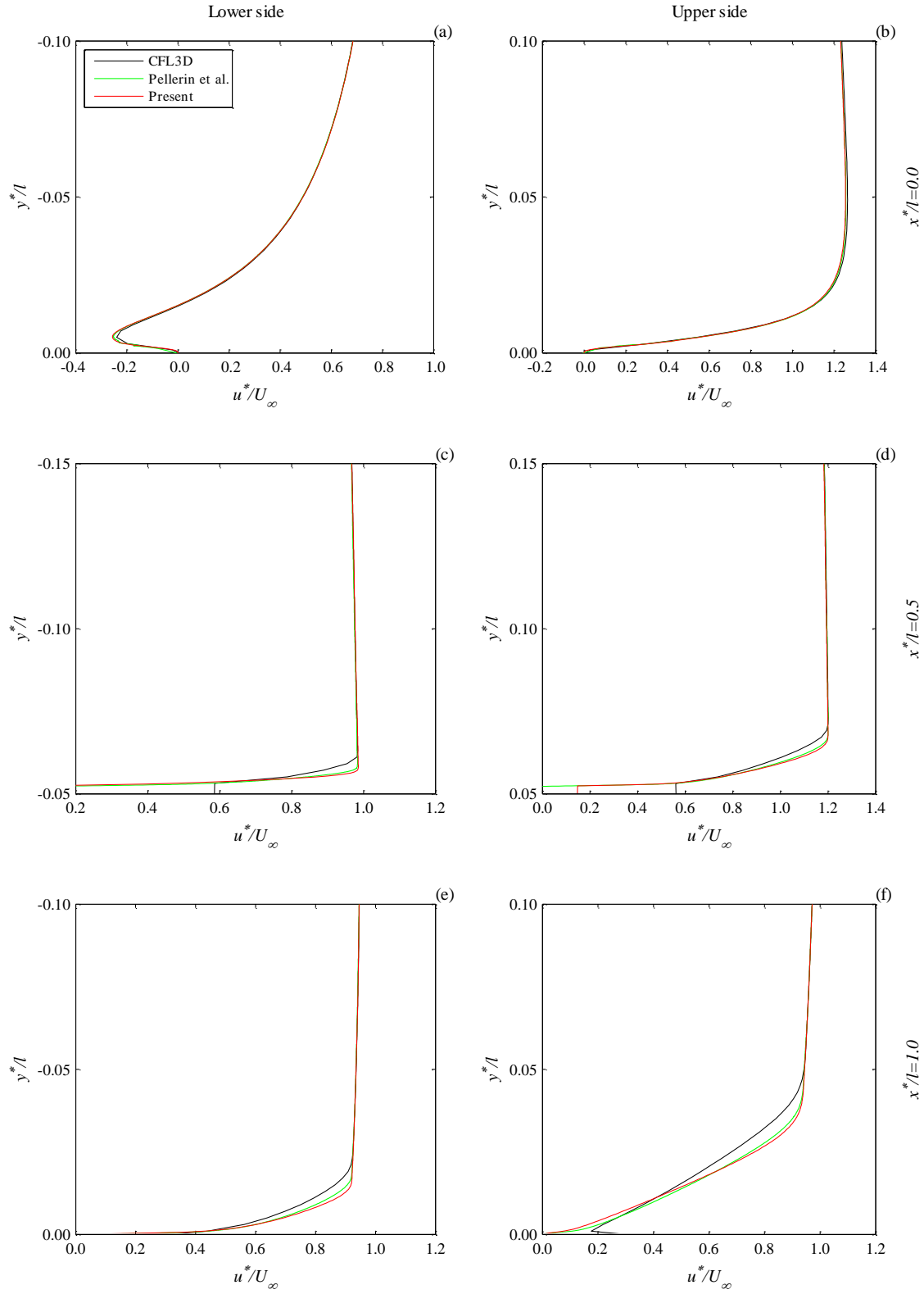


Figure. 12 Velocity profiles for the NACA0012 airfoil at a 7° angle of attack at chord coordinates $x^*/l=0.0$, 0.50 , and 1.0 .

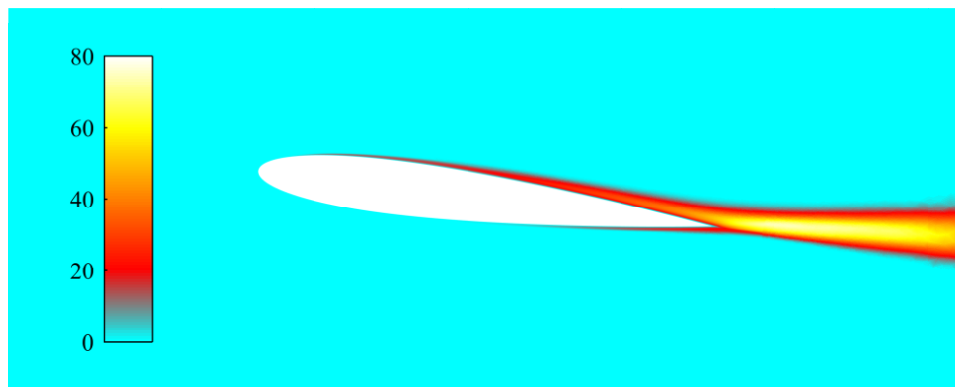


Figure 13: Contour plot of the turbulent viscosity ratio for the NACA0012 airfoil at a 7° angle of attack.

8. Discussion

In this article, we extended the application of the recently developed LBFS to turbulent flows around an airfoil. To do this, we applied a vertex-centered formulation of the method to unstructured meshes and coupled it to a finite-difference implementation of the Spalart-Allmaras turbulence model. Before solving the airfoil flows, the program was validated on standard test cases, without turbulence modeling. First, the lid-driven cavity flow was solved for Reynolds numbers of 100 and 1000 using a Delaunay-type unstructured mesh, the purpose being to verify the accuracy of the method on a mesh that contained low aspect ratio triangles. The resulting centerline velocity profiles and main vortex centers agree well with the data of Ghia et al. (1982), commonly used as a baseline. Second, low Reynolds number steady and unsteady flows around a circular cylinder were investigated, by including larger aspect ratio triangles in the mesh near the cylinder wall, in order to deviate from the Delaunay triangulation near a solid boundary. The program was tested on three meshes with increasing refinement, but in all cases the same aspect ratio was kept for the triangular cells near the wall. The resulting force coefficients and Strouhal numbers show good convergence with mesh refinement. They also compare well with the data selected from other works using finite-volume formulations in conjunction with either the lattice Boltzmann method or a gas kinetic approach.

To test the proposed methodology on turbulent airfoil flows, we constructed unstructured meshes that included larger aspect ratio triangular cells than for the circular cylinder flow test case, especially near the airfoil's mid-chord. The rationale for this mesh design was the necessity of correctly capturing the high-velocity gradients in the wall's normal direction, while limiting the number of vertices in the streamwise direction where the velocity gradients are smaller – see Fig. 8d. Since the LBFS is an explicit unsteady solver, simulations were run until the flow became steady through the effect of the added turbulent viscosity, and the lift and drag coefficients converged to stable values. The resulting lift coefficients compare favorably with the selected reference data obtained with CFL3D, and earlier work of the current authors using a multi-domain LBM solver. The drag coefficients are in excellent agreement with the LBM solver, but still underpredict the CFL3D results slightly. The velocity profiles, too, are in close agreement with their LBM counterparts, and are still fuller than for CFL3D. This demonstrates a consistency between the finite-difference implementation of the Spalart-Allmaras turbulence model in the LBM and LBFS frameworks. In fact, aside from the differences in meshes, the velocity profile

differences with CFL3D can most likely be explained by the choice of turbulence model and its computational implementation. For instance, the CFL3D results were produced with a finite-volume solver and the standard S-A model, whereas the Edwards variant was used in the present work. Also, since the contribution of the wall shear stress to the drag coefficient is larger than that of pressure, and the pressure coefficient profiles that we obtained matched the CFL3D profiles well, it is likely that the small differences in drag are also attributable to the turbulence modeling and its impact on the velocity gradients near the wall. Nevertheless, the purpose of the current work was to test the combination of a turbulence model with an unstructured LBFS, and the results obtained indicate that the proposed method produces valid results. Further simulations could be run with the standard S-A model to verify the impact on the velocity profiles and force coefficients produced, but that is beyond the scope of the current work. For the same reason, no mesh sensitivity analysis was performed, and it is acknowledged that the mesh we constructed may not be optimal, even though its refinement in the wall's normal direction is on a par with that of earlier work (Pellerin et al., 2015) using approximately one quarter the number of vertices. A goal of this work in fact was to use a smaller number of vertices while still providing for an adequate mesh.

While the simulation results that were obtained are satisfactory overall, the present LBFS implementation has two drawbacks. First, the clock time required to achieve steady airfoil flows is high. Even though our Matlab program may not perform computations as quickly as a higher language would, such as C++, the main reason for the long duration is the very small time steps. Near the airfoil wall, the mesh is very fine in the normal direction, and consequently the time steps chosen based on the local LBM lattice stencil size used for computing the cell fluxes scale accordingly. Therefore, even though the flow far from the airfoil stabilizes quickly, especially through the use of local time-stepping, the information travels much more slowly in the vicinity of the airfoil. One remedy for this problem could be the use of wall functions that would limit the normal mesh refinement required. Convergence time improvements could also be sought by using solution adaptive dynamically varying local time steps. The second drawback of the present LBFS derives from its vertex-centered formulation. While this formulation has the advantage of eliminating the need to calculate cell center gradients for calculating boundary fluxes, as in cell-centered formulations, it requires the use of more control surfaces for each control volume. Furthermore, at each control surface, the flux calculations require nine interpolation points for the d2q9 LBM. Overall, this makes the method memory-heavy because interpolation coefficients must be stored for all of these interpolation points.

The LBFS method has a number of advantages. First, the boundary conditions are implemented directly onto the macroscopic quantities, rather than onto the distribution functions as with standard LBM solvers. This simplifies the computational treatment of boundary conditions, since there are no missing post-streaming distributions to be determined. Also, when compared to conventional Navier-Stokes solvers for incompressible flows, the LBFS method does not require the solution of a separate equation for pressure. Moreover, since the density and velocity units used in the macroscopic conservation equations are the same as for the LBM used for the flux calculations, and because the flow physics are, in fact, based on the LBM, the macroscopic pressure is recovered through the $p = \rho c_s^2$ relationship. This was the equation used to calculate the pressure coefficients presented in Figs. 9-10. To recover true macroscopic physical units, appropriate scaling should be applied through the Reynolds number and given free stream density and velocity. A further advantage of the LBFS is its numerical stability. While improving the stability of finite-

volume Navier-Stokes solvers may rely on first or higher order upwind schemes for calculating convective term fluxes, no special scheme is required for the proposed LBFS. This is because the LBM streaming process is by definition applied from an upwind lattice speed direction, and the location of the interpolation points presented in Fig. 1 is chosen such that the local d2q9 stencils cover large areas within their associated triangular cell. Finally, it is interesting to note that, even though the LBFS is based on the BGK collision operator, through the Chapman-Enskog expansion that makes the bridge with the Navier-Stokes equations, it does not suffer from the known instability observed when the relaxation coefficient approaches its maximum value of 2. It is this inherent instability that drove the derivation of more complex LBM collision operators. In fact, for the multi-domain LBM simulation of turbulent airfoil flows, Pellerin et al. (2015) resorted to the cascaded collision operator to maintain numerical stability. In that sense, the LBFS presented here constitutes a simpler computational implementation.

9. Conclusion

In this work, two elements were proposed for extending the applicability of the LBFS. Unstructured meshes that included large aspect ratio triangles for providing directional mesh refinement near walls were used. The method was also coupled to a turbulence model, which made it possible to solve two-dimensional turbulent flows around an airfoil. The results obtained are promising and open the way to possible future validations and developments. For instance, a very interesting test case would be the flow around iced airfoils. Also, validating the method on three-dimensional airfoil flows would broaden its applicability spectrum.

References

- Bhatnagar, P. L., Gross, E. P., & Krook, M. (1954). A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Physical Review*, 94(3), 511–525. <https://doi.org/10.1103/PhysRev.94.511>
- Chen, S., & Doolen, G. D. (1998). Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30, 329–364. <https://doi.org/10.1146/annurev.fluid.30.1.329>
- Filippova, O., Succi, S., Mazzocco, F., Arrighetti, C., Bella, G., Hänel, D., & Dieter, H. (2001). Multiscale lattice Boltzmann schemes with turbulence modeling. *Journal of Computational Physics*, 170(2), 812–829. <https://doi.org/10.1006/jcph.2001.6764>
- Geuzaine, C., & Remacle, J.-F. (2009). Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11), 1309–1331. <https://doi.org/10.1002/nme.2579>
- Ghia, U., Ghia, K. ., & Shin, C. . (1982). High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 48, 387–411. [https://doi.org/10.1016/0021-9991\(82\)90058-4](https://doi.org/10.1016/0021-9991(82)90058-4)
- He, X., & Luo, L.-S. (1997). A priori derivation of the lattice Boltzmann equation. *Physical Review E*, 55(6), R6333–R6336. <https://doi.org/10.1103/PhysRevE.55.R6333>
- Holman, D. M., Brionnaud, R. M., & Modena, M. C. (2014). Aerodynamic analysis of the 2nd high lift prediction workshop by a lattice-Boltzmann method solver. In *32nd AIAA Applied Aerodynamics Conference*. Atlanta, GA: AIAA. <https://doi.org/10.2514/6.2014-2568>

- Imamura, T., Suzuki, K., Nakamura, T., & Yoshida, M. (2005). Flow simulation around an airfoil by lattice Boltzmann method on generalized coordinates. *AIAA Journal*, 43(9), 1968–1973. <https://doi.org/10.2514/1.7554>
- König, B., Fares, E., & Nölting, S. (2014). Lattice-Boltzmann flow simulations for the HiLiftPW-2. In A. SciTech (Ed.), *52nd Aerospace Sciences Meeting* (pp. 1–19). American Institute of Aeronautics and Astronautics. <https://doi.org/10.2514/6.2014-0911>
- Li, K., Zhong, C., Zhuo, C., & Cao, J. (2012). Non-body-fitted Cartesian-mesh simulation of highly turbulent flows using multi-relaxation-time lattice Boltzmann method. *Computers and Mathematics with Applications*, 63(10), 1481–1496. <https://doi.org/10.1016/j.camwa.2012.03.080>
- Li, Y., Shock, R., Zhang, R., Chen, H., & Shih, T. (2005). Simulation of flow over an iced airfoil by using a lattice-Boltzmann method. In *43rd AIAA Aerospace Sciences Meeting and Exhibit* (pp. 1–8). Reno, NV: American Institute of Aeronautics and Astronautics. <https://doi.org/http://dx.doi.org/10.2514/6.2005-1103>
- Lockard, D. P., Luo, L., Milder, S. D., & Singer, B. A. (2002). Evaluation of PowerFLOW for aerodynamic applications. *Journal of Statistical Physics*, 107(April), 423–478.
- Patil, D. V., & Lakshmisha, K. N. (2009). Finite volume TVD formulation of lattice Boltzmann simulation on unstructured mesh. *Journal of Computational Physics*, 228(14), 5262–5279. <https://doi.org/10.1016/j.jcp.2009.04.008>
- Pellerin, N., Leclaire, S., & Reggio, M. (2014). Equilibrium distributions for straight, curved, and immersed boundary conditions in the lattice Boltzmann method. *Computers and Fluids*, (101), 126–135.
- Pellerin, N., Leclaire, S., & Reggio, M. (2015). An implementation of the Spalart–Allmaras turbulence model in a multi-domain lattice Boltzmann method for solving turbulent airfoil flows. *Computers and Mathematics with Applications*, 70(12), 3001–3018. <https://doi.org/10.1016/j.camwa.2015.10.006>
- Peng, G., Xi, H., Duncan, C., & Chou, S.-H. (1998). Lattice Boltzmann method on irregular meshes. *Physical Review E*, 58(4), R4124–R4127. <https://doi.org/10.1103/PhysRevE.58.R4124>
- Peng, G., Xi, H., Duncan, C., & Chou, S.-H. (1999). Finite volume scheme for the lattice Boltzmann method on unstructured meshes. *Physical Review E*, 59(4), 4675–4682. <https://doi.org/10.1103/PhysRevE.59.4675>
- Shu, C., Wang, Y., Teo, C. J., & Wu, J. (2014). Development of lattice Boltzmann flux solver for simulation of incompressible flows. *Advances in Applied Mathematics and Mechanics*, 6(4), 436–460. <https://doi.org/10.4208/aamm.2014.4.s2>
- Stiebler, M., Tölke, J., & Krafczyk, M. (2006). An upwind discretization scheme for the finite volume lattice Boltzmann method. *Computers & Fluids*, 35(8–9), 814–819. <https://doi.org/10.1016/j.compfluid.2005.09.002>
- Sun, Y., Shu, C., Wang, Y., Teo, C. J., & Chen, Z. (2016). An immersed boundary-gas kinetic flux solver for simulation of incompressible flows. *Computers & Fluids*. <https://doi.org/10.1016/j.compfluid.2016.01.007>

- Tian, C. T., Xu, K., Chan, K. L., & Deng, L. C. (2007). A three-dimensional multidimensional gas-kinetic scheme for the Navier-Stokes equations under gravitational fields. *Journal of Computational Physics*, 226(2), 2003–2027. <https://doi.org/10.1016/j.jcp.2007.06.024>
- Viggen, E. M. (2014). *The lattice Boltzmann method: fundamentals and acoustics*. Norwegian University of Science and Technology.
- Wang, Y., Shu, C., Teo, C. J., & Wu, J. (2015). An immersed boundary-lattice Boltzmann flux solver and its applications to fluid-structure interaction problems. *Journal of Fluids and Structures*, 54, 440–465. <https://doi.org/10.1016/j.jfluidstructs.2014.12.003>
- Wang, Y., Shu, C., Teo, C. J., & Yang, L. M. (2016). An efficient immersed boundary-lattice Boltzmann flux solver for simulation of 3D incompressible flows with complex geometry. *Computers & Fluids*, 124, 54–66. <https://doi.org/10.1016/j.compfluid.2015.10.009>
- Wang, Y., Yang, L., & Shu, C. (2015). From lattice Boltzmann method to lattice Boltzmann flux solver. *Entropy*, (17), 7713–7735. <https://doi.org/10.3390/e17117713>
- Yuan, R., Zhong, C., & Zhang, H. (2015). An Immersed-boundary Method Based on the Gas Kinetic BGK Scheme for Incompressible Viscous Flow. *Journal of Computational Physics*, 296(C), 184–208. <https://doi.org/10.1016/j.jcp.2015.04.052>
- Zarghami, A., Maghrebi, M. J., Ghasemi, J., & Ubertini, S. (2012). Lattice Boltzmann finite volume formulation with improved stability. *Communications in Computational Physics*, 12(1), 42–64. <https://doi.org/10.4208/cicp.151210.140711a>

Appendix – Alternate Flux Construction

The alternate approach we use here to construct the flux terms of Eq. (18) is based on the method of Pellerin et al. (2014) for calculating missing post-streaming distributions applicable to boundary conditions. In that work, the missing distributions were approximated by equilibrium distributions streamed from ghost nodes placed at specific distances relative to the nominal lattice grid spacing; more precisely, these ghost nodes were positioned at a distance $\Delta x / (\Delta t / \Delta t_R)$ from the lattice site onto which the distributions were streamed. This process allowed us to avoid approximating the non-equilibrium part of the distributions, and, in turn, enabled us to effectively replace the collide-and-stream sequence with the sole streaming of the equilibrium distributions. The density and velocity necessary for calculating the equilibrium distributions were interpolated or extrapolated at the ghost nodes. This concept is introduced into the LBFS by assuming that the macroscopic interpolation points used in step 1 of the LBFS procedure are the ghost nodes. Steps 1 to 4 remain unchanged. In step 5, the original LBFS method uses Eq. (16) to obtain the non-equilibrium part of the distributions, whereas in our alternate construction, they are simply obtained by subtracting the equilibrium distributions obtained in step 4 from those calculated in step 2. This is because the distributions from step 2 actually replace streamed distributions that include both an equilibrium and a non-equilibrium part. The relaxation coefficient of the BGK collision model is no longer present in the non-equilibrium calculation of step 5. This coefficient only appears in step 6 for calculating Π ; however, it is replaced with $\Delta t^{\text{eff}} / \Delta t_R^{\text{eff}}$, which constitutes an effective relaxation. This change is necessary because the nominal Δx now represents the distance to the ghost node, which makes the effective lattice spacing for the d2q9 stencil, Δx^{eff} , different from Δx . It is

important to point out that Δx^{eff} is only useful in the construction of the alternate model and does not specify any macroscopic interpolation point. It is calculated with the relationship given above as

$$\Delta x^{eff} = \Delta x \frac{\Delta t^{eff}}{\Delta t_R^{eff}} . \quad (A.1)$$

Then, using Eq. (12) and $c = \Delta x / \Delta t = \Delta x^{eff} / \Delta t^{eff} = 1$, the effective relaxation coefficient is obtained from

$$\nu = c_s^2 \left(\Delta t \frac{\Delta t^{eff}}{\Delta t_R^{eff}} \right) \left(\frac{\Delta t_R^{eff}}{\Delta t^{eff}} - \frac{1}{2} \right) , \quad (A.2)$$

which is equivalent to

$$\frac{\Delta t^{eff}}{\Delta t_R^{eff}} = 2 \left(1 - \frac{\nu}{c_s^2 \Delta t} \right) . \quad (A.3)$$

Once the momentum fluxes have been computed in step 6 using the above relaxation coefficient, the macroscopic properties are updated at the vertices through step 8, as in the original LBFS, using the same macroscopic time step value. This time step, which is used in the forward Euler integration, could actually be different and is not strictly restricted to the time step defined in section 2. In fact, it only needs to conform to a CFL condition to maintain numerical stability.

We tested this alternate approach on the NACA0012 flow at 7° in section 7, and obtained identical results to those obtained with the original LBFS. At first this may appear surprising; however, the algebraic development of Eq. (14) in terms of ν , c_s^2 , Δx , and f_i^{eq} following the two approaches yields identical coefficients for $f_i^{eq}(\mathbf{x}, t)$ and $f_i^{eq}(\mathbf{x} - \mathbf{c}_i \Delta t, t - \Delta t)$. Although the method of Pellerin et al. (2014) was developed more empirically, actually it can also be tied to Eq. (15) stemming from the Chapman-Enskog expansion. In fact, the absence of a relaxation coefficient in its finite-difference form, given by Eq. (16) and used in step 5, is equivalent to

$$f_i^{neq}(\mathbf{x}, t) = -\Delta t_R^{eff} \frac{f_i^{eq}(\mathbf{x}, t) - f_i^{eq}(\mathbf{x} - \mathbf{c}_i \Delta t, t - \Delta t)}{\Delta t} , \quad (A.4)$$

where Δt_R^{eff} is set to be equal to Δt . Using this equality, the standard kinematic viscosity-relaxation coefficient relationship of Eq. (12) expressed in terms of the effective time step becomes

$$\nu = c_s^2 \Delta t^{eff} \left(\frac{\Delta t_R^{eff}}{\Delta t^{eff}} - \frac{1}{2} \right) = c_s^2 \Delta t^{eff} \left(\frac{\Delta t}{\Delta t^{eff}} - \frac{1}{2} \right) , \quad (A.5)$$

which implies that

$$\Delta t^{eff} = 2 \left(\Delta t - \frac{\nu}{c_s^2} \right) . \quad (A.6)$$

Therefore,

$$\frac{\Delta t^{eff}}{\Delta t_R^{eff}} = \frac{2 \left(\Delta t - \nu / c_s^2 \right)}{\Delta t} = 2 \left(1 - \frac{\nu}{c_s^2 \Delta t} \right) \quad (A.7)$$

recovers Eq. (A.3), which did not initially originate from Eq. (15). Furthermore, since the lattice speed c is defined as unity regardless of the lattice time step,

$$\frac{\Delta x}{\Delta x^{eff}} = \frac{\Delta t}{\Delta t^{eff}} = \frac{\Delta t_R^{eff}}{\Delta t^{eff}}, \text{ or } \Delta x = \frac{\Delta x^{eff}}{\Delta t^{eff} / \Delta t_R^{eff}}. \quad (\text{A.8})$$

Eq. (A.8) confirms the choice of the ghost node location. There is no loss of generality in this derivation, even though Δt was chosen over Δt^{eff} in Eq. (A.4) for approximating the material derivative of Eq. (15) with a finite difference. The derivative is simply approximated over a shorter time span.

Overall, this alternate flux construction scheme only provides a marginal gain in computational time, as there is one less multiplication performed in step 5 of the LBFS procedure. Nonetheless, we chose to present it here to demonstrate that the applicability of the equilibrium streaming concept proposed in earlier work (Pellerin et al., 2014) is not limited to boundary conditions.